

МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ  
ДЕРЖАВНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД  
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Бессараб В.І.

**МАТЕМАТИЧНІ ОСНОВИ ТЕОРІЇ  
ДИСКРЕТНО-БЕЗПЕРЕРВНИХ СИСТЕМ**

Донецьк  
ДВНЗ «ДонНТУ»  
2011

УДК 681.51.015.4:51

ББК 32.965.6

Рецензенти:

Л.М. Любчик, завідувач кафедри комп'ютерної математики і математичного моделювання Харківського національного технічного університету «ХПІ», д.т.н., професор.

С.В. Мишко, завідувач кафедри прикладної математики і теорії систем управління Донецького національного університету, д.т.н., професор.

Рекомендовано до видання вченою радою ДВНЗ «Донецький національний технічний університет» (Протокол № 9 від 22 грудня 2010 р.)

В.І. Бессараб. Математичні основи теорії дискретно-безперервних систем – Донецьк: ДВНЗ «ДонНТУ», 2011 – 175 с., 42 рис., 5 табл., 1 додаток.

Монографія призначена для фахівців, робота яких пов'язана з розробкою систем автоматизації технологічних процесів. Приведено узагальнені результати досліджень з подальшого розвитку та вдосконалення теорії управління дискретно-безперервними процесами на основі одного з різновидів ідемпотентної алгебри. Сформульовано основні засади аналітичного моделювання дискретно-безперервних систем, наведено практику використання моделей в технічних і технологічних системах на різноманітних прикладах. Детально розглянуто питання синтезу алгоритмів керування дискретно-безперервними об'єктами та специфіку використання відомих методів у відношенні до дискретно-безперервних процесів.

Монографія також може бути корисна для аспірантів і студентів, навчання яких пов'язано з застосуванням теорії автоматичного управління в наукових дослідженнях.

# ЗМІСТ

Вступ.....	5
1 Основи “Max-plus” алгебри.....	8
1.1 Базові операції “Max-plus” алгебри.....	8
1.2 Операції “Max-plus” алгебри з векторами і матрицями.....	10
1.3 Реалізація основних операцій “Max-plus” алгебри за допомогою функцій тулбокса Matlab® .....	14
2 “Max-plus” алгебра і елементи теорії графів.....	23
2.1 Основні поняття і визначення.....	23
2.2 Матриця суміжностей орієнтованого графа в “Max-plus” алгебрі і її властивості.....	27
3 Лінійні рівняння в «Max-plus» алгебрі.....	30
3.1 Рівняння $\mathbf{A} \otimes \mathbf{x} = \mathbf{b}$ і умови його розв’язання.....	30
3.2 Рівняння $\mathbf{x} = \mathbf{A} \otimes \mathbf{x} \oplus \mathbf{b}$ і можливість його розв’язання.....	31
3.3 Спектральне рівняння $\mathbf{A} \otimes \mathbf{x} = \lambda \otimes \mathbf{x}$ .....	33
3.4 Пошук власного вектора.....	35
4 Особливості «Min-plus» алгебри.....	40
5 Основи аналітичного аналізу дискретно-безперервних процесів за допомогою “Max-Plus” алгебри.....	43
6 Приклади аналітичного опису процесів в дискретно- безперервних виробничих системах.....	50
6.1 Виробнича система з паралельно-послідовною обробкою виробів.....	50
6.2 Виробнича система зі змішаною послідовністю обробки виробів.....	53
6.3 Багаторівнева виробнича система.....	58

6.4	Телекомунікаційна мережа з пакетною комутацією.....	63
7	Аналіз асимптотичної поведінки матриці $A^k$ (частотні властивості ДБС).....	72
8	Представлення графа синхронізації ДБС в термінах мережі Петрі за допомогою рівнянь Мах-Plus алгебри.....	80
8.1	Загальний підхід до вирішення задачі.....	80
8.2	Структура і властивості матриці динаміки замкнутої ДБС.....	86
8.3	Практична методика аналізу динаміки ДБС за допомогою “Мах-plus” алгебри.....	88
9	Основи синтезу управління в ДБС.....	99
9.1	Загальний підхід до синтезу управління.....	99
9.2	Структура і властивості матриці динаміки керованої ДБС.....	111
9.3	Аналіз проблеми керування на стартовому циклі.....	112
9.4	Особливості реалізації управління в ДБС з жорсткими логічними умовами.....	117
9.5	Інваріантність в ДБС.....	123
9.6	Компенсація збурень в ДБС.....	127
9.6.1	Компенсація збурень в ДБС з одним власним вектором.....	127
9.6.2	Приклад компенсації збурень в ДБС.....	130
9.6.3	Компенсація збурень в ДБС з декількома власними векторами.....	135
10	Приклад синтезу системи керування хімічним реактором змішування компонентів.....	144
	Висновки.....	153
	Список літератури.....	155
	Додаток.....	164

## Вступ

Математична теорія дискретно-безперервних систем бере початок у роботах відомого американського вченого Ричарда Белмана кінця 50-х років минулого століття і називається “Path-Algebra” або “Dioid” в теорії графів і мереж. Розроблений Белманом вид алгебри дає можливість за допомогою спеціальних математичних рівнянь визначати найкоротшу відстань у телекомунікаційних мережах. На початку 60-х років минулого століття цю ідею підхопив англійський математик Раймонд Канігем-Грін і використовував її під назвою “Minimax-Algebra” для розв’язання широкого кола задач в теорії дослідження операцій. З 80-х років минулого століття ці алгебраїчні методи набувають всезростаючу популярність в теорії керування. Відтоді метод, відомий під назвою “Max-plus” алгебра, успішно застосовується для розв’язання задач автоматизації в галузі транспортних перевезень. Згодом були відкриті нові галузі застосування цього методу, а саме для систем керування об’єктами інших класів [27, 33, 35, 60].

Також в Росії існує відома наукова школа В.П. Маслова, яка займається теоретичними аспектами ідемпотентних алгебр щодо питань оптимального керування дискретно-безперервними процесами.

В останні роки помітний інтерес існує щодо опису динаміки мережевих процесів на основі аналітичних моделей “Max-plus” алгебри [20, 29, 33].

Основна ідея “Max-plus” алгебри полягає в наступному: на деякій множині, що описує стан процесу, встановлюється зв’язок між алгебраїчними операціями. Знаходження максимуму двох елементів позначається символом  $\oplus$  і називається «додавання в “Max-plus” алгебрі». Звичайне додавання двох елементів позначається символом  $\otimes$  і

називається «добуток в “Max-plus” алгебрі». Обидві операції “max” і “plus” складають аббревіатуру “Max-plus” алгебра.

Із системно-теоретичних позицій операція “max” (відповідно  $\oplus$ ) робить можливим опис елементарного процесу синхронізації дискретно-безперервного об’єкта, а за допомогою оператора “plus” (відповідно  $\otimes$ ) при аналізі можна описати розвиток процесу в часі.

Для спрощеного опису деяких особливих випадків “max” і “plus” операцій існують дублюючі оператори:  $\vee$  і  $\wedge$ . Символ  $\vee$  використовують для знаходження мінімуму (“min”) двох елементів, а символ  $\wedge$  застосовують для позначення операції “plus” лише для того, щоб уникнути неоднозначності. Наприклад, для відомої проблеми найкоротшого шляху в телекомунікаційних мережах, де знаходження мінімуму є основною задачею. Структура основних правил і співвідношень при цьому не змінюється. Теорія “Max-plus” алгебри, яка побудована на основі кінцевого набору аксіом, по суті виразів та їх властивостей, подібна до звичайної лінійної алгебри.

Як правило, дискретно-безперервні динамічні системи складаються з елементарних процесів, які відбуваються паралельно і потребують координування. Їх синхронізацію, і отже, повну картину поведінки можливо описати за допомогою рівнянь “Max-plus” алгебри. Алгебраїчне представлення дозволяє провести системний аналіз і надає необхідний базис для формалізованого синтезу систем керування такими процесами.

В останні десятиліття науковий напрямок щодо застосування апарату “Max-plus” алгебри для аналізу і синтезу дискретно-безперервних систем Discrete Event Systems (DESS) розвивається особливо стрімко. Про це свідчить зростаючий потік публікацій в сучасній літературі [36, 38, 40] та увага, що приділяється в програмах наукових конгресів, конференцій і симпозіумів [42, 51, 55]. Це обумовлено різними причинами, але

найважливішою є та, що дискретно-безперервні процеси є досить розповсюдженими в природі, яка нас оточує.

Вагомою допомогою для досліджень у даній області є факт появи спеціалізованого тулбокса [69] для інструментальної системи **Matlab**<sup>®</sup>. Тулбокс “Max-plus” алгебри включає основні обчислювальні функції “Max-plus” алгебри для аналізу і синтезу дискретно-безперервних систем. Всі функції тулбокса представлені у вигляді m-файлів у синтаксисі Matlab<sup>®</sup> і відкриті для користувачів. Також тулбокс є відкритим для включення додаткових функцій користувача, складеними згідно з правилами Matlab<sup>®</sup>.

# 1 Основи “Max-plus” алгебри

## 1.1 Базові операції “Max-plus” алгебри

*Визначення 1.1. “Max-plus” алгебра, найпростіші операції і їх властивості*

“Max-plus” алгебра визначається такими ознаками:

- $R_\varepsilon = R \cup \{-\infty\}$ ,

де  $R$  – множина дійсних чисел;

- $\forall a, b \in R_\varepsilon : a \oplus b = \max(a, b)$ ;

- $\forall a, b \in R_\varepsilon : a \otimes b = a + b$ .

Алгебраїчна множина  $R_{\max} = (R_\varepsilon, \oplus, \otimes)$  називається “Max-plus” алгебра.

При цьому для простоти позначають  $-\infty = \varepsilon$  і  $0 = e$  [35]. Більш точно, алгебраїчна множина  $R_{\max}$  – це ідемпотентний комутативний діюїд, для якого справедливі операції і дії, що наводяться нижче.

Для операції  $\oplus$ :

- асоціативність:

$$\forall a, b, c \in R_\varepsilon : (a \oplus b) \oplus c = a \oplus (b \oplus c);$$

- комутативність:

$$\forall a, b \in R_\varepsilon : a \oplus b = b \oplus a;$$

- існування нейтрального елемента ( $\varepsilon$ ):

$$\forall a \in R_\varepsilon : a \oplus \varepsilon = \varepsilon \oplus a = a .$$

Для операції  $\otimes$ :

- асоціативність:

$$\forall a, b, c \in R_\varepsilon : (a \otimes b) \otimes c = a \otimes (b \otimes c);$$



- існування нейтрального елемента ( $e$ ):  
 $\forall a \in R_\varepsilon : e \otimes a = a \otimes e = a;$
- існування поглинаючого елемента ( $\varepsilon$ ):  
 $\forall a \in R_\varepsilon : a \otimes \varepsilon = \varepsilon;$
- дистрибутивність  $\otimes$  із пріоритетом  $\oplus$  :  
 $\forall a, b, c \in R_\varepsilon : a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c);$   
 $(b \oplus c) \otimes a = (b \otimes a) \oplus (c \otimes a).$

Діюд означає:

- комутативність відносно  $\otimes$  :  
 $\forall a, b \in R_\varepsilon : a \otimes b = b \otimes a;$
- ідемпотентність:  
 $\forall a \in R_\varepsilon : a \oplus a = a.$

Варто підкреслити, що ненульові елементи не мають інверсії відносно  $\oplus$ , тому що  $a \oplus x = b$  не має розв’язку, якщо  $a > b$ .

### Визначення 1.2 Піднесення до степеня в “Max-plus” алгебрі

Припустимо  $a \in R_\varepsilon$  і  $b \in R_\varepsilon$ ,  $a$ -й степінь  $b$  позначається як  $b^a$ , і відповідає операції добутку  $ab$  в звичайній алгебрі.

Якщо  $a = 0$ , тоді  $b^0 = 0$ .

Якщо  $b = \varepsilon$  і  $a > 0$ , тоді  $\varepsilon^a = \varepsilon$ .

Якщо  $b = \varepsilon$  і  $a < 0$ , тоді  $\varepsilon^a$  не визначено.

$\varepsilon^0 = 0$  за визначенням.

### Визначення 1.3 Ділення в “Max-plus” алгебрі

Припустимо  $a \in R$  і  $b \in R_\varepsilon$ , операція ділення визначається в такий спосіб:

$$\frac{a}{b} = a / b = ab^{-1}. \quad (1.1)$$

Якщо  $b = \varepsilon$ , то  $a/b$  не визначено.

З використанням операції ділення можна визначити операцію *min* у такий спосіб:

$$\min(a, b) = ab / (a \oplus b).$$

Добуток  $ab$  в чисельнику, як у звичайній алгебрі.

## 1.2 Операції “Max-plus” алгебри з векторами і матрицями

### Визначення 1.4 Додавання скалярів та векторів

Сумою скалярної величини  $a \in R_\varepsilon$  і вектора  $\mathbf{b} \in R_\varepsilon^n$  (де верхній індекс  $n$  означає розмірність вектора) є вектор  $R_\varepsilon^n$ , що визначається в “Max-plus” алгебрі в такий спосіб:

$$(a \oplus \mathbf{b})_i = a \oplus (\mathbf{b})_i, \quad i = 1, \dots, n. \quad (1.2)$$

### Визначення 1.5 Додавання матриць

Сумою матриць  $\mathbf{A}, \mathbf{B} \in R_\varepsilon^{m \times n}$  (де верхній індекс  $m \times n$  означає розмірність матриці) є матриця  $(\mathbf{A} \oplus \mathbf{B}) \in R_\varepsilon^{m \times n}$ , отримана шляхом додавання відповідних елементів:

$$(\mathbf{A} \oplus \mathbf{B})_{ij} = (\mathbf{A})_{ij} \oplus (\mathbf{B})_{ij}, \quad i = 1, \dots, m, \quad j = 1, \dots, n. \quad (1.3)$$

Додавання скалярних величин і матриць здійснюється тим же чином.

Приклад 1.1 додавання з векторами та матрицями:

$$\text{а) } 4 \oplus \begin{bmatrix} 2 \\ 6 \\ 4 \end{bmatrix} = \begin{bmatrix} 4 \oplus 2 \\ 4 \oplus 6 \\ 4 \oplus 4 \end{bmatrix} = \begin{bmatrix} 4 \\ 6 \\ 4 \end{bmatrix};$$

$$\text{б) } 5 \oplus \begin{bmatrix} 9 & 6 & 3 \\ 7 & 4 & 1 \\ 3 & 1 & 5 \end{bmatrix} = \begin{bmatrix} 5 \oplus 9 & 5 \oplus 6 & 5 \oplus 3 \\ 5 \oplus 7 & 5 \oplus 4 & 5 \oplus 1 \\ 5 \oplus 3 & 5 \oplus 1 & 5 \oplus 5 \end{bmatrix} = \begin{bmatrix} 9 & 6 & 5 \\ 7 & 5 & 5 \\ 5 & 5 & 5 \end{bmatrix};$$

$$\text{в) } \begin{bmatrix} \varepsilon & 4 \\ 3 & 5 \\ 8 & 1 \end{bmatrix} \oplus \begin{bmatrix} 3 & 5 \\ 2 & 6 \\ 3 & e \end{bmatrix} = \begin{bmatrix} \varepsilon \oplus 3 & 4 \oplus 5 \\ 3 \oplus 2 & 5 \oplus 6 \\ 8 \oplus 3 & 1 \oplus e \end{bmatrix} = \begin{bmatrix} 3 & 5 \\ 3 & 6 \\ 8 & 1 \end{bmatrix}.$$

*Визначення 1.6 Добуток вектора і скалярної величини*

Добуток скалярної величини  $a \in R_\varepsilon$  і вектора  $\mathbf{b} \in R_\varepsilon^n$  :

$$(\mathbf{a} \otimes \mathbf{b})_i = a \otimes b_i, \quad i = 1, \dots, n. \quad (1.4)$$

*Визначення 1.7 Добуток матриць*

Добуток матриць  $\mathbf{A} \in R_\varepsilon^{m \times p}$  і  $\mathbf{B} \in R_\varepsilon^{p \times n}$  є матрицею  $(\mathbf{A} \otimes \mathbf{B}) \in R_\varepsilon^{m \times n}$ , кожний елемент якої є сумою добутків відповідних елементів  $i$ -го рядку  $\mathbf{A}$  і  $j$ -го стовпця  $\mathbf{B}$  :

$$(\mathbf{A} \otimes \mathbf{B})_{ij} = \bigoplus_{k=1}^p (\mathbf{A})_{ik} \otimes (\mathbf{B})_{kj} = \max_k ((\mathbf{A})_{ik} + (\mathbf{B})_{kj}), \quad i = 1, \dots, m, \quad j = 1, \dots, n, \quad (1.5)$$

де  $\bigoplus_{j=1}^m a_j$  – короткий запис  $a_1 \oplus a_2 \oplus \dots \oplus a_m$ .

Добутки скалярної величини і матриці, а також вектора і матриці визначаються так само.

Приклад 1.2 обчислення добутків:

$$\text{а) } 3 \otimes \begin{bmatrix} 4 \\ 8 \\ 12 \end{bmatrix} = \begin{bmatrix} 3 \otimes 4 \\ 3 \otimes 8 \\ 3 \otimes 12 \end{bmatrix} = \begin{bmatrix} 7 \\ 11 \\ 15 \end{bmatrix};$$

$$\text{б) } [2 \quad 4 \quad 6] \otimes \begin{bmatrix} 2 & 4 & e \\ 7 & \varepsilon & -1 \\ e & 5 & \varepsilon \end{bmatrix} =$$

$$\begin{aligned}
 &= [(2 \otimes 2) \oplus (4 \otimes 7) \oplus (6 \otimes e); (2 \otimes 4) \oplus (4 \otimes \varepsilon) \oplus (6 \otimes 5); \\
 &\quad (2 \otimes e) \oplus (4 \otimes (-1)) \oplus (6 \otimes \varepsilon)] = \\
 &= [4 \oplus 11 \oplus 6; 6 \oplus \varepsilon \oplus 11; 2 \oplus 3 \oplus \varepsilon] = [11 \ 11 \ 3];
 \end{aligned}$$

$$\begin{aligned}
 \text{в)} \quad & \begin{bmatrix} 1 & e & e \\ e & 5 & \varepsilon \end{bmatrix} \otimes \begin{bmatrix} 7 & 9 \\ 6 & 4 \\ 2 & 3 \end{bmatrix} = \\
 &= \begin{bmatrix} (1 \otimes 7) \oplus (e \otimes 6) \oplus (e \otimes 2) & (1 \otimes 9) \oplus (e \otimes 4) \oplus (e \otimes 3) \\ (e \otimes 7) \oplus (5 \otimes 6) \oplus (\varepsilon \otimes 2) & (e \otimes 9) \oplus (5 \otimes 4) \oplus (\varepsilon \otimes 3) \end{bmatrix} = \\
 &= \begin{bmatrix} 8 \oplus 6 \oplus 2 & 10 \oplus 4 \oplus 3 \\ 7 \oplus 11 \oplus \varepsilon & 9 \oplus 9 \oplus \varepsilon \end{bmatrix} = \begin{bmatrix} 8 & 10 \\ 11 & 9 \end{bmatrix}.
 \end{aligned}$$

#### Визначення 1.8 Нейтральна матриця

Матриця  $\mathbf{I}_n \in R^{n \times n}$  з елементами  $e$  на головній діагоналі і іншими елементами  $\varepsilon$  називається нейтральною матрицею  $n$ -го порядку.

#### Визначення 1.9 Нульова матриця

Матриця  $\boldsymbol{\varepsilon}^{m \times n}$  з елементами  $\varepsilon_{ij} = \varepsilon$  для всіх  $i, j$ , називається нульовою матрицею.

#### Визначення 1.10 Піднесення матриці до степені

Нехай  $\mathbf{A} \in R_\varepsilon^{n \times m}$  і  $b \in N_0$ , тоді  $b$ -а степінь  $\mathbf{A}$  визначається:

$$\mathbf{A}^b = \underbrace{\mathbf{A} \otimes \dots \otimes \mathbf{A}}_b = \bigotimes_{i=1}^b \mathbf{A}, \quad (1.6)$$

Якщо  $b = 0$ , тоді  $\mathbf{A}^0 = \mathbf{I}$ .

Якщо  $b = -1$ , дивися визначення 1.11.

Якщо  $b \notin N_0 \cup \{-1\}$  тоді  $\mathbf{A}^b$  не визначено, де  $N_0$  – набір невід'ємних цілих чисел.

*Визначення 1.11 Обчислення зворотної матриці, умова оборотності*

Матриця  $\mathbf{A} \in R_{\varepsilon}^{n \times n}$  називається оборотною в “Max-plus” алгебрі, якщо її можна представити у вигляді добутку:  $\mathbf{A} = \mathbf{D} \otimes \mathbf{P}$ , де:

$\mathbf{D} \in R_{\varepsilon}^{n \times n}$  – матриця, на діагоналі якої немає жодного  $\varepsilon$  елемента,

$\mathbf{P} \in R_{\varepsilon}^{n \times n}$  – матриця перестановки.

Тоді

$$\mathbf{A}^{-1} = \mathbf{P}^{-1} \otimes \mathbf{D}^{-1}, \quad (1.7)$$

де:

$$(\mathbf{D}^{-1})_{ii} = -(\mathbf{D})_{ii},$$

$$\mathbf{P}^{-1} = \mathbf{P}^T.$$

Приклад 1.3 Інверсія матриці

Розглянемо  $\mathbf{A} = \begin{bmatrix} \varepsilon & 5 & \varepsilon \\ \varepsilon & \varepsilon & e \\ -1 & \varepsilon & \varepsilon \end{bmatrix}$ .

Ця матриця оборотна:  $\mathbf{A} = \mathbf{D} \otimes \mathbf{P}$ , де  $\mathbf{D} = \begin{bmatrix} 5 & \varepsilon & \varepsilon \\ \varepsilon & e & \varepsilon \\ \varepsilon & \varepsilon & -1 \end{bmatrix}$  і  $\mathbf{P} = \begin{bmatrix} \varepsilon & e & \varepsilon \\ \varepsilon & \varepsilon & e \\ e & \varepsilon & \varepsilon \end{bmatrix}$ .

Маємо  $\mathbf{D}^{-1} = \begin{bmatrix} -5 & \varepsilon & \varepsilon \\ \varepsilon & e & \varepsilon \\ \varepsilon & \varepsilon & 1 \end{bmatrix}$  і  $\mathbf{P}^{-1} = \begin{bmatrix} \varepsilon & \varepsilon & e \\ e & \varepsilon & \varepsilon \\ \varepsilon & e & \varepsilon \end{bmatrix}$ , отже,

$$\mathbf{A}^{-1} = \mathbf{P}^{-1} \otimes \mathbf{D}^{-1} = \begin{bmatrix} \varepsilon & \varepsilon & 1 \\ -5 & \varepsilon & \varepsilon \\ \varepsilon & e & \varepsilon \end{bmatrix}.$$

*Визначення 1.12 Ділення матриць в “Max-plus” алгебрі*

Припустимо  $\mathbf{A} \in R_{\varepsilon}^{n \times n}$  і  $\mathbf{B} \in R_{\varepsilon}^{n \times n}$ . Якщо  $\mathbf{B}$  оборотна, то ділення  $\mathbf{A}$  на  $\mathbf{B}$  визначається:

$$\mathbf{A} / \mathbf{B} = \mathbf{B}^{-1} \otimes \mathbf{A}. \quad (1.8)$$

Визначення 1.13 Слід матриці

Слід матриці  $\mathbf{A} \in R_{\varepsilon}^{n \times n}$  – це сума елементів головної діагоналі, що позначається  $tr(\mathbf{A})$ :

$$tr(\mathbf{A}) = \bigoplus_{i=1}^n (\mathbf{A})_{ii} \quad (1.9)$$

Еквівалентно,  $tr(\mathbf{A})$  – це максимальний елемент головної діагоналі.

### 1.3 Реалізація основних операцій “Max-plus” алгебри за допомогою функцій тулбокса Matlab<sup>®</sup>

**Функції завдання даних:**

• *mp\_zeros* – нульовий масив (скаляр, вектор або матриця)

```
>> mp_zeros
```

```
ans =
```

```
-Inf
```

```
>> mp_zeros(5,6)
```

```
ans =
```

```
-Inf -Inf -Inf -Inf -Inf -Inf
```

```
-Inf -Inf -Inf -Inf -Inf -Inf
```

```
-Inf -Inf -Inf -Inf -Inf -Inf
```

```
-Inf -Inf -Inf -Inf -Inf -Inf
```

```
-Inf -Inf -Inf -Inf -Inf -Inf
```

• *mp\_ones* – одиничний масив (скаляр, вектор або матриця)

```
>> mp_ones
```

ans =

0

>> mp\_ones(4,6)

ans =

0 0 0 0 0 0

0 0 0 0 0 0

0 0 0 0 0 0

0 0 0 0 0 0

• ***mp\_eye(n,m)*** *n*-на-*m* нейтральна матриця ( $\mathbf{I}_n$ )

>> mp\_eye

ans =

0

>> mp\_eye(3,4)

ans =

0 -Inf -Inf -Inf

-Inf 0 -Inf -Inf

-Inf -Inf 0 -Inf

### Функції обчислень

• ***mp\_sum(A,B)*** – сума  $\mathbf{A}$  і  $\mathbf{B}$  ( $\mathbf{A} \oplus \mathbf{B}$ )

1. Припустимо необхідно знайти суму числа  $\mathbf{A} = 4$  і  $\mathbf{B} = \begin{bmatrix} 2 \\ 6 \\ 4 \end{bmatrix}$ :

$$4 \oplus \begin{bmatrix} 2 \\ 6 \\ 4 \end{bmatrix} = \begin{bmatrix} 4 \oplus 2 \\ 4 \oplus 6 \\ 4 \oplus 4 \end{bmatrix} = \begin{bmatrix} 4 \\ 6 \\ 4 \end{bmatrix}.$$

Цю же суму можна знайти, використовуючи функцію “Max-plus” алгебри  $mp\_sum(A,B)$ :

```
>> mp_sum(4,[2; 6; 4])
```

```
ans =
```

```
4
6
4
```

2. Припустимо необхідно знайти суму двох матриць:  $\mathbf{A} = \begin{bmatrix} \varepsilon & 4 \\ 3 & 5 \\ 8 & 1 \end{bmatrix}$  і

$$\mathbf{B} = \begin{bmatrix} 3 & 5 \\ 2 & 6 \\ 3 & e \end{bmatrix}:$$

$$\begin{bmatrix} \varepsilon & 4 \\ 3 & 5 \\ 8 & 1 \end{bmatrix} \oplus \begin{bmatrix} 3 & 5 \\ 2 & 6 \\ 3 & e \end{bmatrix} = \begin{bmatrix} \varepsilon \oplus 3 & 4 \oplus 5 \\ 3 \oplus 2 & 5 \oplus 6 \\ 8 \oplus 3 & 1 \oplus e \end{bmatrix} = \begin{bmatrix} 3 & 5 \\ 3 & 6 \\ 8 & 1 \end{bmatrix}.$$

```
>> A=[mp_zeros 4; 3 5; 8 1], B=[3 5; 2 6; 3 mp_ones]
```

```
A =
```

```
-Inf 4
3 5
8 1
```

```
B =
```

```
3 5
2 6
```



3 0

>> mp\_sum(A,B)

ans =

3 5

3 6

8 1

• **mp\_multi(A,B)** добуток **A** і **B** ( $A \otimes B$ )

1. Припустимо необхідно знайти добуток числа  $A = 3$  і  $B = \begin{bmatrix} 4 \\ 8 \\ 12 \end{bmatrix}$ :

$$3 \otimes \begin{bmatrix} 4 \\ 8 \\ 12 \end{bmatrix} = \begin{bmatrix} 3 \otimes 4 \\ 3 \otimes 8 \\ 3 \otimes 12 \end{bmatrix} = \begin{bmatrix} 7 \\ 11 \\ 15 \end{bmatrix}.$$

Цей же добуток можна знайти, використавши функцію “Max-plus” алгебри  $mp\_multi(A,B)$ :

>> mp\_multi(3, [4; 8; 12])

ans =

7

11

15

2. Припустимо необхідно знайти добуток  $A = [2 \ 4 \ 6]$  і

$$B = \begin{bmatrix} 2 & 4 & e \\ 7 & \varepsilon & -1 \\ e & 5 & \varepsilon \end{bmatrix}:$$

$$\begin{aligned}
 [2 \ 4 \ 6] \otimes \begin{bmatrix} 2 & 4 & e \\ 7 & \varepsilon & -1 \\ e & 5 & \varepsilon \end{bmatrix} &= \\
 &= [(2 \otimes 2) \oplus (4 \otimes 7) \oplus (6 \otimes e); \quad (2 \otimes 4) \oplus (4 \otimes \varepsilon) \oplus (6 \otimes 5); \\
 &\quad (2 \otimes e) \oplus (4 \otimes (-1)) \oplus (6 \otimes \varepsilon)] = \\
 &= [4 \oplus 11 \oplus 6; \quad 6 \oplus \varepsilon \oplus 11; \quad 2 \oplus 3 \oplus \varepsilon] = [11 \ 11 \ 3].
 \end{aligned}$$

>> A=[2 4 6]

A =

2 4 6

>> B=[2 4 mp\_ones; 7 mp\_zeros -1; mp\_ones 5 mp\_zeros]

B =

2 4 0

7 -Inf -1

0 5 -Inf

>> mp\_multi(A,B)

ans =

11 11 3

3. Припустимо необхідно знайти добуток  $\mathbf{A} = \begin{bmatrix} 1 & e & e \\ e & 5 & \varepsilon \end{bmatrix}$  і  $\mathbf{B} = \begin{bmatrix} 7 & 9 \\ 6 & 4 \\ 2 & 3 \end{bmatrix}$ :

$$\begin{aligned}
 \begin{bmatrix} 1 & e & e \\ e & 5 & \varepsilon \end{bmatrix} \otimes \begin{bmatrix} 7 & 9 \\ 6 & 4 \\ 2 & 3 \end{bmatrix} &= \begin{bmatrix} (1 \otimes 7) \oplus (e \otimes 6) \oplus (e \otimes 2) & (1 \otimes 9) \oplus (e \otimes 4) \oplus (e \otimes 3) \\ (e \otimes 7) \oplus (5 \otimes 6) \oplus (\varepsilon \otimes 2) & (e \otimes 9) \oplus (5 \otimes 4) \oplus (\varepsilon \otimes 3) \end{bmatrix} = \\
 &= \begin{bmatrix} 8 \oplus 6 \oplus 2 & 10 \oplus 4 \oplus 3 \\ 7 \oplus 11 \oplus \varepsilon & 9 \oplus 9 \oplus \varepsilon \end{bmatrix} = \begin{bmatrix} 8 & 10 \\ 11 & 9 \end{bmatrix}.
 \end{aligned}$$

```
>> A=[1 mp_ones mp_ones; mp_ones 5 mp_zeros]
```

```
A =
```

```
1 0 0
0 5 -Inf
```

```
>> B=[7 9; 6 4; 2 3]
```

```
B =
```

```
7 9
6 4
2 3
```

```
>> mp_multi(A,B)
```

```
ans =
```

```
8 10
11 9
```

•  $mp\_power(A,n)$   $n$ -а степені  $A$  ( $A^n$ )

Припустимо необхідно піднести до 3 степені матрицю  $A = \begin{bmatrix} 1 & 5 \\ 9 & 4 \end{bmatrix}$ :

$$\begin{aligned} \begin{bmatrix} 1 & 5 \\ 9 & 4 \end{bmatrix}^3 &= \left( \begin{bmatrix} 1 & 5 \\ 9 & 4 \end{bmatrix} \otimes \begin{bmatrix} 1 & 5 \\ 9 & 4 \end{bmatrix} \right) \otimes \begin{bmatrix} 1 & 5 \\ 9 & 4 \end{bmatrix} = \\ &= \begin{bmatrix} (1 \otimes 1) \oplus (5 \otimes 9) & (1 \otimes 5) \oplus (5 \otimes 4) \\ (9 \otimes 1) \oplus (4 \otimes 9) & (9 \otimes 5) \oplus (4 \otimes 4) \end{bmatrix} \otimes \begin{bmatrix} 1 & 5 \\ 9 & 4 \end{bmatrix} = \\ &= \begin{bmatrix} 14 & 9 \\ 13 & 14 \end{bmatrix} \otimes \begin{bmatrix} 1 & 5 \\ 9 & 4 \end{bmatrix} = \begin{bmatrix} (14 \otimes 1) \oplus (9 \otimes 9) & (14 \otimes 5) \oplus (9 \otimes 4) \\ (13 \otimes 1) \oplus (14 \otimes 9) & (13 \otimes 5) \oplus (14 \otimes 4) \end{bmatrix} = \begin{bmatrix} 18 & 19 \\ 23 & 18 \end{bmatrix}. \end{aligned}$$

Піднесення до степені можна зробити використавши функцію "Max-plus" алгебри  $mp\_power(A,n)$

```
>> mp_power([1 5; 9 4],3)
```

ans =

18 19

23 18

•  $mp\_inv(A)$  зворотна матриця  $\mathbf{A}$  ( $\mathbf{A}^{-1}$ )

Припустимо, нам необхідно знайти матрицю, зворотну до матриці  $\mathbf{A}$ .

Розглянемо  $\mathbf{A} = \begin{bmatrix} \varepsilon & 5 & \varepsilon \\ \varepsilon & \varepsilon & e \\ -1 & \varepsilon & \varepsilon \end{bmatrix}$ . Ця матриця оборотна:  $\mathbf{A} = \mathbf{D} \otimes \mathbf{P}$ , де

$\mathbf{D} = \begin{bmatrix} 5 & \varepsilon & \varepsilon \\ \varepsilon & e & \varepsilon \\ \varepsilon & \varepsilon & -1 \end{bmatrix}$  і  $\mathbf{P} = \begin{bmatrix} \varepsilon & e & \varepsilon \\ \varepsilon & \varepsilon & e \\ e & \varepsilon & \varepsilon \end{bmatrix}$ . Маємо  $\mathbf{D}^{-1} = \begin{bmatrix} -5 & \varepsilon & \varepsilon \\ \varepsilon & e & \varepsilon \\ \varepsilon & \varepsilon & 1 \end{bmatrix}$  і

$\mathbf{P}^{-1} = \begin{bmatrix} \varepsilon & \varepsilon & e \\ e & \varepsilon & \varepsilon \\ \varepsilon & e & \varepsilon \end{bmatrix}$ , отже  $\mathbf{A}^{-1} = \mathbf{P}^{-1} \otimes \mathbf{D}^{-1} = \begin{bmatrix} \varepsilon & \varepsilon & 1 \\ -5 & \varepsilon & \varepsilon \\ \varepsilon & e & \varepsilon \end{bmatrix}$ .

```
>> A=[-Inf 5 -Inf; -Inf -Inf 0; -1 -Inf -Inf]
```

A =

```
-Inf 5 -Inf
```

```
-Inf -Inf 0
```

```
-1 -Inf -Inf
```

```
>> mp_inv(A)
```

ans =

```
-Inf -Inf 1
```

```
-5 -Inf -Inf
```

```
-Inf 0 -Inf
```

•  $mp\_div(A,B)$  ділення  $A$  на  $B$  ( $B^{-1} \otimes A$ )

$$\text{Припустимо } B = \begin{bmatrix} \varepsilon & 5 & \varepsilon \\ \varepsilon & \varepsilon & e \\ -1 & \varepsilon & \varepsilon \end{bmatrix}, \text{ тоді } B^{-1} = P^{-1} \otimes D^{-1} = \begin{bmatrix} \varepsilon & \varepsilon & 1 \\ -5 & \varepsilon & \varepsilon \\ \varepsilon & e & \varepsilon \end{bmatrix} \quad (3)$$

попереднього прикладу). Припустимо  $A = \begin{bmatrix} 3 & 6 & 3 \\ 4 & -2 & \varepsilon \\ 1 & 0 & 8 \end{bmatrix}$ .

$$A/B = B^{-1} \otimes A = \begin{bmatrix} \varepsilon & \varepsilon & 1 \\ -5 & \varepsilon & \varepsilon \\ \varepsilon & e & \varepsilon \end{bmatrix} \otimes \begin{bmatrix} 3 & 6 & 3 \\ 4 & -2 & \varepsilon \\ 1 & e & 8 \end{bmatrix} = \begin{bmatrix} 2 & 1 & 9 \\ -2 & 1 & -2 \\ 4 & -2 & \varepsilon \end{bmatrix}.$$

```
>> A=[3 6 3; 4 -2 -Inf; 1 0 8]
```

```
A =
```

```
3 6 3
4 -2 -Inf
1 0 8
```

```
>> B=[-Inf -Inf 3; -2 -Inf -Inf; -Inf 0 -Inf]
```

```
B =
```

```
-Inf 5 -Inf
-Inf -Inf 0
-1 -Inf -Inf
```

```
>> mp_div(A,B)
```

```
ans =
```

```
2 1 9
-2 1 -2
4 -2 -Inf
```

•  $mp\_trace(A)$  слід  $A$  ( $tr(A)$  )

```
>> A=[3 6 3; 4 -2 -Inf; 1 0 8]
```

```
A =
```

```
 3   6   3
```

```
 4  -2 -Inf
```

```
 1   0   8
```

```
>> mp_trace(A)
```

```
ans =
```

```
 8
```

Приведені функції основних операцій “Max-plus” алгебри демонструють можливість використання спеціалізованого тулбокса. Більш повний опис його можливостей наведено в [69].

## 2 “Max-plus” алгебра і елементи теорії графів

Термінологія сучасної теорії графів досить різноманітна і на сьогоднішній день остаточно не встановилася. В цьому розділі розглянуто основні поняття і визначення теорії графів, з використанням яких можна зручно і наочно описувати дискретно-безперервні процеси. В сукупності з використанням базових операцій “Max-plus” алгебри такий підхід дозволяє розв’язувати задачі аналізу на графових моделях в новій інтерпретації.

### 2.1 Основні поняття і визначення

#### *Визначення 2.1 Поняття графа і диграфа*

Спрямований граф  $G$  (або скорочено диграф) визначається як пара  $(\mathbf{v}, \eta)$ , де  $\mathbf{v}$  – набір елементів, які називаються вузлами (або вершинами) графа, а  $\eta$  – набір елементів, за допомогою яких з’єднуються пари вузлів, тобто ребра графа.

#### *Визначення 2.2 Зважений диграф*

Спрямований граф називається зваженим, якщо існує елемент, пов’язаний з кожним ребром  $(i, j) \in \eta$ , чисельне значення якого називається вагою ребра  $(i, j)$  і позначається  $a_{ij}$ .

#### *Визначення 2.3 Початковий і кінцевий вузол, джерело і приймач*

Номери окремих вузлів прийнято позначати як  $1, 2, \dots, n$ . Якщо  $(i, j) \in \eta$ , то  $i$  називається початковим вузлом або коренем ребра  $(i, j)$ , а  $j$  – кінцевим вузлом або кінцевою точкою ребра  $(i, j)$ . В деяких випадках використовують наступну термінологію. Для  $(i, j) \in \eta$ ,  $i$  називається

джерелом для  $j$ , а  $j$  називається приймачем для  $i$ . Набір всіх джерел для  $j$  позначають  $\pi(j)$ .

*Визначення 2.4 Шлях, контур і ациклічний диграф*

Шлях  $\rho$  у графі – це послідовність вузлів  $(i_1, i_2, \dots, i_p)$ ,  $p > 1$ , така що  $i_k \in \pi(i_{k+1})$ ,  $k = 1, \dots, p-1$ . Іншими словами, шлях – це послідовність ребер, що з'єднує послідовність вузлів графа. Контур у графі – це шлях, в якому збігаються початковий і кінцевий вузли послідовності. Спрямований граф називається ациклічним, якщо він не містить контурів.

*Визначення 2.5 Довжина і вага шляху або контуру*

Довжина шляху або контуру позначається  $|\rho|_l$  та дорівнює числу ребер, з яких він складається. Вага шляху або контуру позначається  $|\rho|_w$  та дорівнює сумі ваг ребер, що утворюють шлях або контур.

*Визначення 2.6 Зв'язний і жорстко зв'язний диграф*

Диграф називається зв'язним графом, якщо існує шлях між деякими парами вершин.

Спрямований граф називається жорстко зв'язним, якщо існує шлях між всіма парами його вершин. Приклад зв'язного (але не жорстко) спрямованого графа наведено на рисунку 2.1.

$$\mathbf{A} = \begin{bmatrix} -\infty & -\infty & -\infty & -\infty \\ 3 & -\infty & -\infty & -\infty \\ -\infty & 5 & -\infty & -\infty \\ 5 & 6 & -\infty & 4 \end{bmatrix};$$

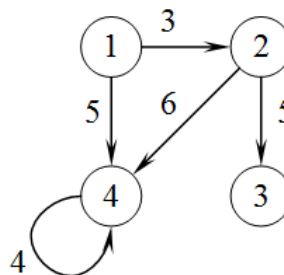


Рисунок 2.1 – Зв'язний (але не жорстко) диграф



Нехай  $\mathbf{A} \in R_\varepsilon^{n \times n}$  і  $\mathbf{B}$  матриця:

$$\mathbf{B} = \mathbf{A} \oplus \mathbf{A}^2 \oplus \dots \oplus \mathbf{A}^n, \quad (2.1)$$

тоді  $G(\mathbf{A})$  жорстко зв'язний, якщо кожний елемент  $\mathbf{B}$  більше  $\varepsilon$ .

*Визначення 2.7 Пріоритетний граф*

Пріоритетний граф  $G(\mathbf{A})$  матриці  $\mathbf{A} \in R_\varepsilon^{n \times n}$  – це зважений диграф з  $n$  вузлами і ребрами  $(i, j)$  у тому випадку, якщо  $(A)_{ij} \neq \varepsilon$ .

а)

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 & -\infty \\ 3 & 2 & -\infty & -\infty \\ 4 & -\infty & 3 & 12 \\ -\infty & -\infty & 7 & 4 \end{bmatrix};$$

б)

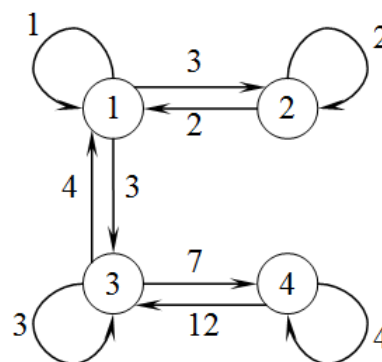


Рисунок 2.2 – Приклад матриці (а) і її пріоритетного графа (б)

*Визначення 2.8 Максимальне значення контуру*

Максимальне значення контуру пріоритетного графа  $G(\mathbf{A})$  – це максимум середніх ваг контурів.

$$\lambda = \max_{\rho} \frac{|\rho|_{\omega}}{|\rho|_l}, \quad (2.2)$$

де  $|\rho|_l$  – довжина контуру,  $|\rho|_{\omega}$  – вага контуру.

Інакше максимальне значення контуру визначається таким способом:

$$\lambda_i = \bigoplus_{j=1}^n \left( \text{tr}(\mathbf{A}^j)^{1/j} \right), \quad (2.3)$$

де  $\mathbf{A}^j$  – максимальна вага всіх контурів довжиною  $j$  по вузлах  $i$ ,  $i = 1, \dots, n$ .

$tr(\mathbf{A}^j)$  – максимум за всіма вузлами.

Якщо пріоритетний граф  $G(\mathbf{A})$  жорстко зв'язний, то

$$\lambda = \max_{\substack{1 \leq j \leq n \\ (\mathbf{A}^n)_{ij} \neq -\infty}} \min_{1 \leq k \leq n} \frac{(\mathbf{A}^n)_{ij} - (\mathbf{A}^{n-k})_{ij}}{k}. \quad (2.4)$$

В позначеннях “Max-plus”:

$$\lambda = \bigoplus_{\substack{1 \leq j \leq n \\ (\mathbf{A}^n)_{ij} \neq -\infty}} \bigvee_{1 \leq k \leq n} \left( \frac{(\mathbf{A}^n)_{ij}}{(\mathbf{A}^{n-k})_{ij}} \right)^{\frac{1}{k}} \quad (2.5)$$

де:

$$a \vee b \equiv \min(a, b).$$

Приклад 2.1 Обчислимо максимальне значення контуру  $\lambda$  для графа, зображеного на рисунку 2.2. Для цього складемо таблицю, у яку запишемо вагу всіх шляхів довжини  $j$ , що проходять через вузол  $i$  і обчислимо  $(\mathbf{A}_{ij})$ :

**Таблиця 2.1** – Обчислення  $\lambda$  для графа

вузол	шлях довжини	вага шляху	$(\mathbf{A}_{ij})$
$i = 1$	$j = 1$	1	1
	$j = 2$	5, 7	7
	$j = 3$	6, 7, 8, 10	10
	$j = 4$	8, 11, 26	26
$i = 2$	$j = 1$	2	2
	$j = 2$	5	5
	$j = 3$	6, 7	7
	$j = 4$	8, 12	12

Продовження таблиці 2.1

вузол	шлях довжини	вага шляху	$(\mathbf{A}_{ij})$
$i = 3$	$j = 1$	3	3
	$j = 2$	7, 19	19
	$j = 3$	8, 10, 23, 22	23
	$j = 4$	12, 11, 26, 26	26
$i = 4$	$j = 1$	4	4
	$j = 2$	19	19
	$j = 3$	23, 22	23
	$j = 4$	26, 26	26

Обчислимо  $tr(A^j)$  і  $(tr(A^j))^{1/j}$ . Оскільки  $tr(\mathbf{A}^j) = \bigoplus_{j=1}^4 (\mathbf{A}_{ij})$ , одержимо:

$$tr(A^1) = 1 \oplus 2 \oplus 3 \oplus 4 = 4, \quad (4)^{1/1} = 4;$$

$$tr(A^2) = 7 \oplus 5 \oplus 19 \oplus 19 = 19, \quad (19)^{1/2} = 9.5;$$

$$tr(A^3) = 10 \oplus 7 \oplus 23 \oplus 23 = 23, \quad (23)^{1/3} = 7.67;$$

$$tr(A^4) = 26 \oplus 12 \oplus 26 \oplus 26 = 26, \quad (26)^{1/4} = 6.5.$$

Отже, згідно з (2.3),  $\lambda = 4 \oplus 9.5 \oplus 7.67 \oplus 6.5 = 9.5$ .

## 2.2 Матриця суміжності орієнтованого графа в “Max-plus” алгебрі і її властивості

Розглянемо довільну квадратну матрицю  $\mathbf{A} = (a_{ij}) \in R_{\varepsilon}^{n \times n}$ , як матрицю суміжності деякого орієнтованого графа. Елемент матриці  $a_{ij} \neq \varepsilon$  вказує на існування дуги  $(j, i)$  в графі, а  $a_{ij} = \varepsilon$  – на її відсутність.

Обчислимо матрицю  $\mathbf{A}^2 = \mathbf{A} \otimes \mathbf{A}$  і позначимо через  $a_{ij}^{(2)}$  її елементи. Елемент цієї матриці  $a_{ij}^{(2)} \neq \varepsilon$  тоді і тільки тоді, коли існує принаймні один

шлях з вузла  $j$  у вузол  $i$ , що містить дві дуги. Аналогічно, для будь-якого цілого  $k > 0$ , матриця  $\mathbf{A}^k$  має елемент  $a_{ij}^{(k)} \neq \varepsilon$  тоді і тільки тоді, коли існує шлях довжини  $k$  з вузла  $j$  у вузол  $i$ . Дійсно, оскільки

$$a_{ij}^{(k)} = \bigoplus_{i_1, \dots, i_{(k-1)}=1}^n a_{ii_1} \otimes a_{i_1 i_2} \otimes \dots \otimes a_{i_{(k-1)} j} \neq \varepsilon,$$

то це означає, що існує послідовність індексів  $\{i_1, \dots, i_{(k-1)}\}$ , для яких всі елементи  $a_{ii_1}, a_{i_1 i_2}, \dots, a_{i_{(k-1)} j}$  для матриці  $\mathbf{A}$  не дорівнюють  $\varepsilon$ . Звідси витікає, що існує шлях з вузла  $j$  до вузла  $i$  довжиною  $k$  через вузли, які відповідають виділеним індексам.

Якщо граф, що відповідає матриці  $\mathbf{A}$  є ациклічним, то з вище представлених міркувань витікає, що  $\mathbf{A}^k = \varepsilon$  для кожного  $k > r$ , де  $r$  – довжина найбільшого шляху в графі. У протилежному випадку, якщо граф, що відповідає матриці  $\mathbf{A}$ , має замкнутий контур, то  $\mathbf{A}^k \neq \varepsilon$  для всіх  $k = 1, 2, \dots$

Приклад 2.2. Розглянемо дві матриці на рис. 2.3 і рис. 2.4, які є матрицями суміжностей. Поруч з ними зображені орієнтовані графи. Граф на рис. 2.3 не має замкнутих контурів і відповідна йому матриця суміжності ациклічна. Довжина найбільшого шляху в даному графі дорівнює 3. Тому має виконуватися умова  $\mathbf{A}^k = \varepsilon$  для всіх  $k > 3$ , що дійсно має місце. У другому ж графі є цикл  $\{v_1, v_2, v_4\}$ , з цього витікає, що  $\mathbf{A}^k \neq \varepsilon$  для всіх  $k = 1, 2, \dots$

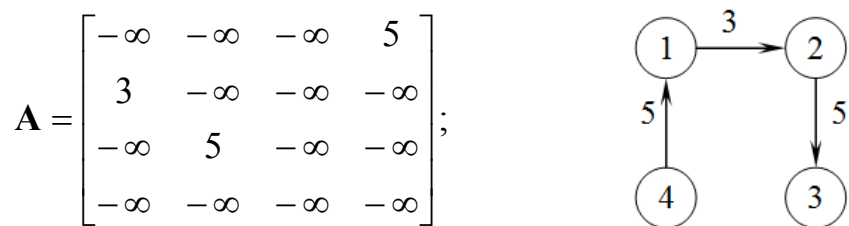


Рисунок 2.3 – Ациклічний диграф

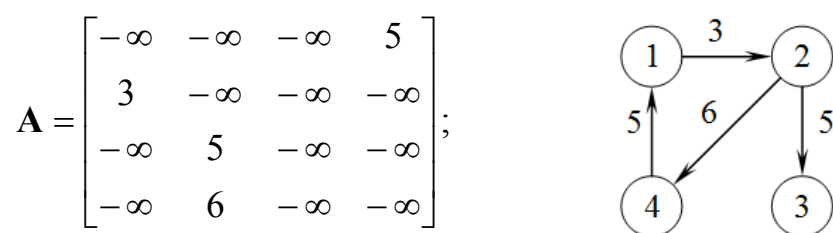


Рисунок 2.4 – Граф з циклом  $\{v_1, v_2, v_4\}$

### 3 Лінійні рівняння в «Max-plus» алгебрі

#### 3.1 Рівняння $\mathbf{A} \otimes \mathbf{x} = \mathbf{b}$ і умови його розв'язання

«Max-plus» алгебра має таку властивість:

$$\forall a, b \in R_\varepsilon : a \leq b \Leftrightarrow a \oplus b = b. \quad (3.1)$$

Розглянемо такий вираз. Припустимо  $\mathbf{A} \in R_\varepsilon^{n \times n}$  і  $\mathbf{b} \in R_\varepsilon^n$ :

$$\mathbf{A} \otimes \mathbf{x} = \mathbf{b} \quad (3.2)$$

В загальному випадку рівняння (3.2) не має розв'язку, але якщо виконується нерівність у вигляді:

$$\mathbf{A} \otimes \mathbf{x} \leq \mathbf{b} \quad (3.3)$$

то в цьому випадку розв'язок існує і визначається як:

$$(\mathbf{i} \oslash \mathbf{x})^T = (\mathbf{i} \oslash \mathbf{b})^T \otimes \mathbf{A}, \quad (3.4)$$

де  $\mathbf{i}$  – Max-plus алгебраїчний вектор ідентичності відповідної розмірності.

Приклад 3.1.

$$\text{Припустимо } \mathbf{A} = \begin{bmatrix} \varepsilon & \varepsilon & 6 & \varepsilon \\ 3 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & 5 & \varepsilon & \varepsilon \\ 5 & 6 & \varepsilon & 4 \end{bmatrix}, \mathbf{b} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}.$$

Знайдемо розв'язок рівняння  $\mathbf{A} \otimes \mathbf{x} = \mathbf{b}$ .

$$\text{Запишемо вектор ідентичності } \mathbf{i} = \begin{bmatrix} e \\ e \\ e \\ e \end{bmatrix}. \quad \text{Тоді } \mathbf{i} \oslash \mathbf{b} = \begin{bmatrix} -1 \\ -2 \\ -3 \\ -4 \end{bmatrix},$$

$$(\mathbf{i} \oslash \mathbf{b})^T = [-1 \quad -2 \quad -3 \quad -4].$$

Знайдемо добуток  $(i0\mathbf{b})^T \otimes \mathbf{A}$ :

$$(i0\mathbf{b})^T \otimes \mathbf{A} = [-1 \quad -2 \quad -3 \quad -4] \otimes \begin{bmatrix} \varepsilon & \varepsilon & 6 & \varepsilon \\ 3 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & 5 & \varepsilon & \varepsilon \\ 5 & 6 & \varepsilon & 4 \end{bmatrix} = [1 \quad 2 \quad 5 \quad e].$$

Згідно з (3.4)  $(i0\mathbf{x})^T = (i0\mathbf{x})^T \otimes \mathbf{A}$ . Таким чином,  $(i0\mathbf{x})^T = [1 \quad 2 \quad 5 \quad e]$ .

З цього витікає, що  $\mathbf{x}^T = [-1 \quad -2 \quad -5 \quad e]$ .

### 3.2 Рівняння $\mathbf{x} = \mathbf{A} \otimes \mathbf{x} \oplus \mathbf{b}$ і можливість його розв'язання

Припустимо  $\mathbf{A} \in R_{\varepsilon}^{n \times n}$  і  $\mathbf{b} \in R_{\varepsilon}^n$ . Будемо розглядати неявне відносно  $\mathbf{x} = (x_1, \dots, x_n)^T$  рівняння такого виду:

$$\mathbf{x} = \mathbf{A} \otimes \mathbf{x} \oplus \mathbf{b}. \quad (3.5)$$

Знання умов можливості розв'язання цього рівняння відносно  $\mathbf{x}$  має велике значення при дослідженні динамічних процесів в дискретно-безперервних системах. Умови, за яких існує явний розв'язок (3.5) і його вид можуть бути сформульовані в такий спосіб.

Нехай елементи матриці  $\mathbf{A}$  і вектора  $\mathbf{b}$  або додатні, або дорівнюють  $\varepsilon$ . Тоді рівняння (3.5) має єдиний розв'язок  $\mathbf{x} \in R_{\varepsilon}^n$  тоді і тільки тоді, коли граф, що відповідає матриці  $\mathbf{A}$ , є ациклічним. В тому випадку, коли розв'язок існує, він представляється у вигляді:

$$\mathbf{x} = (\mathbf{I} \oplus \mathbf{A})^r \otimes \mathbf{b}, \quad (3.6)$$

де  $r$  – довжина найбільшого шляху в графі.

Для доказу цього твердження почнемо з підстановки  $\mathbf{x}$  з рівняння (3.5) в його праву частину. Зробимо таку операцію  $k$  раз:

$$\mathbf{x} = \mathbf{A} \otimes \mathbf{x} \oplus \mathbf{b} = \mathbf{A}^{(k+1)} \otimes \mathbf{x} \oplus (\mathbf{I} \oplus \mathbf{A} \oplus \dots \oplus \mathbf{A}^k) \otimes \mathbf{b},$$

де для матриці  $\mathbf{A}$ , що відповідає ациклічному графу (і тільки для такої матриці)  $\mathbf{A}^{(k+1)} = \varepsilon$  для всіх  $k \geq r$ . Враховуючи те, що

$$(\mathbf{I} \oplus \mathbf{A} \oplus \dots \oplus \mathbf{A}^k) = (\mathbf{I} \otimes \mathbf{A})^k$$

одержимо умову твердження. Таким чином, розв'язок рівняння визначається рівністю

$$\mathbf{x} = \mathbf{A}^* \otimes \mathbf{b}. \quad (3.7)$$

*Визначення 3.1 Оператор  $*$ .*

Оператор  $*$  для квадратної матриці  $\mathbf{A} \in R_{\varepsilon}^{n \times n}$  визначається як:

$$\mathbf{A}^* = \bigoplus_{k \in \mathbf{N}_0} \mathbf{A}^k, \quad (3.8)$$

де  $\mathbf{A}^0 = \mathbf{I}_n$ ,  $\mathbf{A}^k = \mathbf{A} \otimes \mathbf{A}^{k-1}$  і  $\mathbf{N}_0$  – набір невід'ємних цілих чисел.

Можемо інтерпретувати  $(\mathbf{A}^*)_{ij}$  як максимальну вагу шляху від вузла  $i$  до вузла  $j$  довільної довжини в графі  $G(\mathbf{A})$ . При цьому  $\mathbf{A}^*$  визначається в  $(R_{\varepsilon} \cup \{+\infty\})^{n \times n}$ , але значення  $+\infty$  не має змісту в більшості практичних додатків [46].

Припустимо  $\mathbf{A} \in R_{\varepsilon}^{n \times n}$ . Якщо немає контурів з додатною вагою в пріоритетному графі  $G(\mathbf{A})$ , тоді

$$\mathbf{A}^* = \mathbf{A}^0 \oplus \mathbf{A}^1 \oplus \dots \oplus \mathbf{A}^{n-1} \quad (3.9)$$

Приклад 3.2.

$$\text{Нехай } \mathbf{A} = \begin{bmatrix} -1 & -2 & -3 & \varepsilon \\ -3 & -2 & \varepsilon & \varepsilon \\ -4 & \varepsilon & -3 & -12 \\ \varepsilon & \varepsilon & -7 & -4 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} -1 \\ 2 \\ 3 \\ \varepsilon \end{bmatrix}.$$

Знайдемо розв'язок рівняння  $\mathbf{x} = \mathbf{A} \otimes \mathbf{x} \oplus \mathbf{b}$ .



Обчислимо  $\mathbf{A}^*$ . Оскільки в пріоритетному графі, що відповідає матриці  $\mathbf{A}$ , немає контурів з додатною вагою і порядок матриці  $n=4$ , маємо:

$$\mathbf{A}^* = \mathbf{I} \oplus \mathbf{A} \oplus \mathbf{A}^2 \oplus \mathbf{A}^3,$$

$$\mathbf{A}^* = \begin{bmatrix} e & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & e & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & e & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & e \end{bmatrix} \oplus \begin{bmatrix} -1 & -2 & -3 & \varepsilon \\ -3 & -2 & \varepsilon & \varepsilon \\ -4 & \varepsilon & -3 & -12 \\ \varepsilon & \varepsilon & -7 & -4 \end{bmatrix} \oplus \begin{bmatrix} -2 & -3 & -4 & -15 \\ -4 & -4 & -6 & \varepsilon \\ -5 & -6 & -6 & -15 \\ -11 & \varepsilon & -10 & -8 \end{bmatrix} \oplus$$

$$\oplus \begin{bmatrix} -3 & -4 & -5 & -16 \\ -5 & -6 & -7 & -18 \\ -6 & -7 & -8 & -18 \\ -12 & -13 & -13 & -12 \end{bmatrix} = \begin{bmatrix} e & -2 & -3 & -15 \\ -3 & e & -6 & -18 \\ -4 & -6 & e & -12 \\ -11 & -13 & -7 & e \end{bmatrix}.$$

Отже, розв'язок рівняння:

$$\mathbf{x} = \mathbf{A}^* \otimes \mathbf{b} = \begin{bmatrix} e & -2 & -3 & -15 \\ -3 & e & -6 & -18 \\ -4 & -6 & e & -12 \\ -11 & -13 & -7 & e \end{bmatrix} \otimes \begin{bmatrix} -1 \\ 2 \\ 3 \\ \varepsilon \end{bmatrix} = \begin{bmatrix} e \\ 2 \\ 3 \\ -4 \end{bmatrix}.$$

### 3.3 Спектральне рівняння $\mathbf{A} \otimes \mathbf{x} = \lambda \otimes \mathbf{x}$

Матриця  $\mathbf{A} \in R_\varepsilon^{n \times n}$  має власне число в Max-plus алгебрі, якщо існують дійсне число  $\lambda \in R$  і вектор  $\mathbf{v} \in R^n$  такі, що

$$\mathbf{A} \otimes \mathbf{v} = \lambda \otimes \mathbf{v}. \quad (3.10)$$

Вектор  $\mathbf{v}$  називають власним вектором для власного числа  $\lambda$ . Ця теорія дуже схожа на відому теорію Перона-Фробеніуса [35]. Кожна квадратна матриця з елементами в  $R_\varepsilon$  має хоча б одне власне число. Відмінністю в Max-plus алгебрі є те, що матриця може мати декілька (не пропорційних) власних векторів.

Розв'язок рівняння (3.10) можливий, тільки якщо  $G(\mathbf{A})$  містить замкнутий контур.

Якщо  $G(\mathbf{A})$  жорстко зв'язний граф з одним замкнутим контуром, то існує одне і тільки одне власне число (але, можливо, кілька власних векторів). Це власне число еквівалентне максимальному значенню ваги контура  $G(\mathbf{A})$ .

Якщо  $G(\mathbf{A})$  не жорстко зв'язний, але в ньому існують замкнуті контури, то також можна знайти контур з максимальною вагою  $G(\mathbf{A})$ , яку можна трактувати як власне число відповідної матриці  $\mathbf{A}$  [71].

### Приклад 3.3

Приклад, що ілюструє наявність єдиного власного значення, якщо  $G(\mathbf{A})$  не замкнений, але має такі контури:

$$\begin{bmatrix} 2 & \varepsilon \\ \varepsilon & 2 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ \varepsilon \end{bmatrix} = 2 \otimes \begin{bmatrix} 1 \\ \varepsilon \end{bmatrix} = \begin{bmatrix} 3 \\ \varepsilon \end{bmatrix},$$

$$\begin{bmatrix} 2 & \varepsilon \\ \varepsilon & 2 \end{bmatrix} \otimes \begin{bmatrix} \varepsilon \\ 1 \end{bmatrix} = 2 \otimes \begin{bmatrix} \varepsilon \\ 1 \end{bmatrix} = \begin{bmatrix} \varepsilon \\ 3 \end{bmatrix}.$$

Приклад замкненого, але не строго, графа  $G(\mathbf{A})$ , для якого існує два власних значення:

$$\begin{bmatrix} 1 & 1 \\ \varepsilon & 2 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ \varepsilon \end{bmatrix} = 1 \otimes \begin{bmatrix} 1 \\ \varepsilon \end{bmatrix} = \begin{bmatrix} 2 \\ \varepsilon \end{bmatrix},$$

$$\begin{bmatrix} 1 & 1 \\ \varepsilon & 2 \end{bmatrix} \otimes \begin{bmatrix} \varepsilon \\ 1 \end{bmatrix} = 2 \otimes \begin{bmatrix} \varepsilon \\ 1 \end{bmatrix} = \begin{bmatrix} \varepsilon \\ 3 \end{bmatrix}.$$

Існує інший спосіб одержання власного значення  $\lambda$  з використанням початкових умов. Початкова умова – це лінійна система (3.11), що описує поведінку в часі автономної ДБС.

$$\mathbf{x}(k) = \mathbf{A} \otimes \mathbf{x}(k-1), \quad (3.11)$$

де  $\mathbf{x} \in R_\varepsilon^n$  і  $\mathbf{A} \in R_\varepsilon^{n \times n}$ .

Якщо задано початковий вектор  $\mathbf{x}(0)$  (початковий стан системи), то говорять, що система (3.11) після певного числа кроків приходить до кінцевого стану, у разі якщо існують цілі числа  $p, q: p > q \geq 0$  і дійсне число  $c$  таке, що  $\mathbf{x}(p) = \mathbf{x}(q) \otimes c$ . Таким чином, власне число  $\lambda$  визначається:

$$\lambda = \frac{c}{p - q} \quad (\text{як у звичайній алгебрі}) \quad (3.12)$$

Приклад 3.4

Припустимо  $\mathbf{x}(0) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$  і  $\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 & \varepsilon \\ 3 & 2 & \varepsilon & \varepsilon \\ 4 & \varepsilon & 3 & 12 \\ \varepsilon & \varepsilon & 7 & 4 \end{bmatrix}$ . Тоді

$$\mathbf{x}(0) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \rightarrow \mathbf{x}(1) = \begin{bmatrix} 3 \\ 3 \\ 12 \\ 17 \end{bmatrix} \rightarrow \mathbf{x}(2) = \begin{bmatrix} 15 \\ 6 \\ 19 \\ 19 \end{bmatrix} \rightarrow \mathbf{x}(3) = \begin{bmatrix} 22 \\ 18 \\ 31 \\ 26 \end{bmatrix} \rightarrow \mathbf{x}(4) = \begin{bmatrix} 34 \\ 25 \\ 38 \\ 38 \end{bmatrix}.$$

Неважко бачити, що  $\mathbf{x}(4) = \mathbf{x}(2) \otimes 19$ . Отже,  $p = 4$ ,  $q = 2$ ,  $c = 19$ . А власне число:

$$\lambda = \frac{19}{4 - 2} = \frac{19}{2} = 9.5.$$

### 3.4 Пошук власного вектора

#### Алгоритм 3.1 [70]

Дано  $p$ ,  $q$  і  $c$  (див. співвідношення 3.12), тоді власний вектор:

$$\mathbf{v} = \bigoplus_{j=1}^{p-q} (\lambda^{p-q-j} \otimes \mathbf{x}(q + j - 1)). \quad (3.13)$$

Приклад 3.5

Обчислимо власний вектор за алгоритмом 3.1.

$$\text{Припустимо } \mathbf{x}(0) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \text{ і } \mathbf{A} = \begin{bmatrix} 1 & 2 & 3 & \varepsilon \\ 3 & 2 & \varepsilon & \varepsilon \\ 4 & \varepsilon & 3 & 12 \\ \varepsilon & \varepsilon & 7 & 4 \end{bmatrix}. \text{ Відомо, що } p=4, q=2,$$

$c=19$  і  $\lambda=9.5$ . Відповідно до алгоритму  $\mathbf{v} = \bigoplus_{j=1}^2 (\lambda^{2-j} \otimes \mathbf{x}(j+1))$ .

$$\text{Отже, } \mathbf{v} = \lambda^1 \otimes \mathbf{x}(2) \oplus \lambda^0 \otimes \mathbf{x}(3) = 9.5 \otimes \begin{bmatrix} 15 \\ 6 \\ 19 \\ 19 \end{bmatrix} \oplus 1 \otimes \begin{bmatrix} 22 \\ 18 \\ 31 \\ 26 \end{bmatrix} = \begin{bmatrix} 24.5 \\ 18 \\ 31 \\ 28.5 \end{bmatrix}.$$

**Алгоритм 3.2** (власний вектор-кандидат) – так званий перший силовий алгоритм [70]

Якщо відомо  $p, q$  і  $c$ , тоді власний вектор:

$$\mathbf{v} = \frac{1}{p-q} \sum_{j=1}^{p-q} \mathbf{x}(q+j-1) \text{ (як у звичайній алгебрі)} \quad (3.14)$$

Зауважимо: цей алгоритм не завжди дає власний вектор.

Приклад 3.6

$$\text{Припустимо } \mathbf{x}(0) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \text{ і } \mathbf{A} = \begin{bmatrix} 1 & 2 & 3 & \varepsilon \\ 3 & 2 & \varepsilon & \varepsilon \\ 4 & \varepsilon & 3 & 12 \\ \varepsilon & \varepsilon & 7 & 4 \end{bmatrix}. \text{ Обчислимо власний}$$

вектор за алгоритмом 3.2. Відомо, що  $p=4, q=2, c=19$  і  $\lambda=9.5$ .

Відповідно до алгоритму  $\mathbf{v} = \frac{1}{2} \sum_{j=1}^2 \mathbf{x}(j+1)$ .

$$\text{Отже, } \mathbf{v} = \frac{1}{2}[\mathbf{x}(2) + \mathbf{x}(3)] = \frac{1}{2} \cdot \left( \begin{bmatrix} 15 \\ 6 \\ 19 \\ 19 \end{bmatrix} + \begin{bmatrix} 22 \\ 18 \\ 31 \\ 26 \end{bmatrix} \right) = \frac{1}{2} \cdot \begin{bmatrix} 37 \\ 24 \\ 50 \\ 45 \end{bmatrix} = \begin{bmatrix} 18.5 \\ 12 \\ 25 \\ 22.5 \end{bmatrix}.$$

Зауважимо, що власні вектора, отримані за алгоритмами 3.1 і 3.2 лінійно-залежні з коефіцієнтом 6.

### Алгоритм 3.3 (другий силовий алгоритм) [70]

1. Обчислення кандидата на власний вектор за (3.14).
2. Якщо  $\mathbf{v}$  – власний вектор, то обчислення закінчено. Інакше перейти до кроку 3.
3. Визначити новий вектор  $\bar{\mathbf{v}}$ :

$$\bar{v}_i = \begin{cases} v_i, & \text{якщо } (\mathbf{A} \otimes \mathbf{v})_i = \lambda \otimes v_i \\ \varepsilon, & \text{інакше} \end{cases}$$

4. Прийняти, що  $\mathbf{x}(0) = \bar{\mathbf{v}}$  і для  $r \geq 0$  проводити рекурентні розрахунки поки не буде виконана умова:  $\mathbf{x}(r+1) = \lambda \otimes \mathbf{x}(r)$ . Тоді  $\mathbf{x}(r)$  і є власним вектором для  $\mathbf{A}$ .

#### Приклад 3.7

$$\text{Припустимо } \mathbf{x}(0) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \varepsilon \end{bmatrix} \text{ і } \mathbf{A} = \begin{bmatrix} 0 & 4 & \varepsilon & \varepsilon \\ -1 & \varepsilon & \varepsilon & 3 \\ \varepsilon & \varepsilon & 2 & \varepsilon \\ \varepsilon & 1 & -7 & \varepsilon \end{bmatrix}. \text{ Відомо, що } \lambda = 2.$$

Обчислимо власний вектор за алгоритмом 3.3.

Результат за алгоритмом 3.2:  $\mathbf{v} = \begin{bmatrix} 4 \\ 1.5 \\ 3 \\ 0.5 \end{bmatrix}$ . Легко бачити, що

$$\mathbf{A} \otimes \mathbf{v} = \begin{bmatrix} 5.5 \\ 3.5 \\ 5 \\ 2.5 \end{bmatrix} \neq \begin{bmatrix} 6 \\ 3.5 \\ 5 \\ 2.5 \end{bmatrix} = \lambda \otimes \mathbf{v}. \text{ Обчислимо новий вектор } \bar{\mathbf{v}} = \begin{bmatrix} \varepsilon \\ 1.5 \\ 3 \\ 0.5 \end{bmatrix}.$$

Якщо  $\mathbf{x}(0) = \bar{\mathbf{v}}$ , тоді

$$\mathbf{x}(0) = \begin{bmatrix} \varepsilon \\ 1.5 \\ 3 \\ 0.5 \end{bmatrix} \rightarrow \mathbf{x}(1) = \begin{bmatrix} 5.5 \\ 3.5 \\ 5 \\ 2.5 \end{bmatrix} \rightarrow \mathbf{x}(2) = \begin{bmatrix} 7.5 \\ 5.5 \\ 7 \\ 4.5 \end{bmatrix} = \lambda \otimes \mathbf{x}(1).$$

Отже, власний вектор дорівнює  $\begin{bmatrix} 5.5 \\ 3.5 \\ 5 \\ 2.5 \end{bmatrix}$ .

### Алгоритм 3.4 (Викладено в [63])

1. Обчислення кандидата на власний вектор за (3.14).
2. Якщо  $\mathbf{v}$  – власний вектор, то обчислення закінчено, інакше перейти до кроку 3.
3. Перерахувати алгоритм з  $\mathbf{x}(0) = \mathbf{v}$ , доки для  $r \geq 0$  не виконається рівняння  $\mathbf{x}(r+1) = \lambda \otimes \mathbf{x}(r)$ , тоді  $\mathbf{x}(r)$  буде власним вектором для  $\mathbf{A}$  і обчислення закінчено.

### Приклад 3.8

Припустимо  $\mathbf{x}(0) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \varepsilon \end{bmatrix}$  і  $\mathbf{A} = \begin{bmatrix} 0 & 4 & \varepsilon & \varepsilon \\ -1 & \varepsilon & \varepsilon & 3 \\ \varepsilon & \varepsilon & 2 & \varepsilon \\ \varepsilon & 1 & -7 & \varepsilon \end{bmatrix}$ . Відомо, що  $\lambda = 2$ .

Обчислимо власний вектор за алгоритмом 3.4.

Результат за алгоритмом (3.2):  $\mathbf{v} = \begin{bmatrix} 4 \\ 1.5 \\ 3 \\ 0.5 \end{bmatrix}$ . Легко бачити, що

$$\mathbf{A} \otimes \mathbf{v} = \begin{bmatrix} 5.5 \\ 3.5 \\ 5 \\ 2.5 \end{bmatrix} \neq \begin{bmatrix} 6 \\ 3.5 \\ 5 \\ 2.5 \end{bmatrix} = \lambda \otimes \mathbf{v}. \text{ Складемо новий вектор } \bar{\mathbf{v}} = \begin{bmatrix} 4 \\ 1.5 \\ 3 \\ 0.5 \end{bmatrix}.$$

Якщо  $\mathbf{x}(0) = \bar{\mathbf{v}}$ , тоді

$$\mathbf{x}(0) = \begin{bmatrix} 4 \\ 1.5 \\ 3 \\ 0.5 \end{bmatrix} \rightarrow \mathbf{x}(1) = \begin{bmatrix} 5.5 \\ 3.5 \\ 5 \\ 2.5 \end{bmatrix} \rightarrow \mathbf{x}(2) = \begin{bmatrix} 7.5 \\ 5.5 \\ 7 \\ 4.5 \end{bmatrix} = \lambda \otimes \mathbf{x}(1).$$

Отже, власний вектор дорівнює  $\begin{bmatrix} 5.5 \\ 3.5 \\ 5 \\ 2.5 \end{bmatrix}$ .

## 4 Особливості «Min-plus» алгебри

Визначення 4.1 «Min-plus» алгебра

«Min-plus» алгебра визначається такими ознаками

- $R_\varepsilon = R \cup \{+\infty\}$ ,
- де  $R$  – множина дійсних чисел;
- $\forall a, b \in R_\varepsilon : a \oplus b = \min(a, b)$ ;
- $\forall a, b \in R_\varepsilon : a \otimes b = a + b$

Алгебраїчна множина  $R_{\min} = (R_\varepsilon, \oplus, \otimes)$  називається “Min-plus” алгебра.

Операції мінімуму в Min-plus алгебрі і максимуму в Max-plus алгебрі є тотожними, тому  $R_{\min}$  і  $R_{\max}$  – ізоморфні алгебраїчні структури. Але, щоб уникнути двозначності, надалі будемо використовувати різні символи:

$$a \vee b \equiv \min(a, b);$$

$$a \wedge b \equiv a + b.$$

Ми використовуємо різні символи для операції  $+$  в Max-plus і Min-plus, оскільки

$$-\infty \otimes \infty = -\infty \neq -\infty \wedge \infty = \infty. \quad (4.1)$$

Max-plus і Min-plus алгебра разом складають *min-max алгебру* [71].

Основні скалярні, векторні і матричні операції в Min-plus формуються аналогічно операціям “Max-plus”.

Визначення 4.2. Оператор  $\nabla$ .

Для квадратної матриці  $\mathbf{A} \in R_\varepsilon^{n \times n}$ :

$$\mathbf{A}^\nabla = \bigvee_{k \in \mathbf{N}} \mathbf{A}^k, \quad (4.2)$$

де  $\mathbf{N}$  – множина натуральних чисел.



Це означає, що

$$\mathbf{A}^\nabla \vee \mathbf{A}^0 = \mathbf{A}^* \quad (4.3)$$

Матрицю  $\mathbf{A}^\nabla$  іноді називають матрицею найкоротшого шляху.

Приклад 4.1.

Знайдемо найкоротший шлях між 1-м і 6-м вузлом у графі, зображеному на рисунку 4.1.

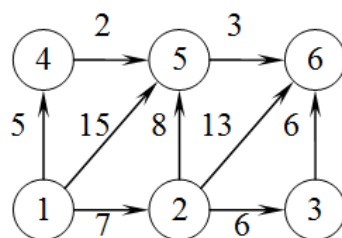


Рисунок 4.1 – Диграф

Випишемо матрицю графа

$$\mathbf{A} = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 7 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & 6 & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 5 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 15 & 8 & \varepsilon & 2 & \varepsilon & \varepsilon \\ \varepsilon & 13 & 6 & \varepsilon & 3 & \varepsilon \end{bmatrix}.$$

Тоді

$$\mathbf{A}^\nabla = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 7 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 13 & 6 & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 5 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 7 & 8 & \varepsilon & 2 & \varepsilon & \varepsilon \\ 10 & 11 & 6 & 5 & 3 & \varepsilon \end{bmatrix}.$$

Оскільки  $(\mathbf{A}^\nabla)_{6,1} = 10$ , то найкоротший шлях від вузла 1 до вузла 6 дорівнює 10.

Порівняємо результат, отриманий засобами “Min-plus” алгебри з результатами роботи алгоритму Дейкстри і алгоритму Флойда. Випишемо таблицю для алгоритму Дейкстри.

**Таблиця 4.1 – Алгоритм Дейкстри**

	1	2	3	4	5	6
крок1	[0]	7	–	5	15	–
крок2	[0]	7	–	[5]	7	–
крок3	[0]	[7]	13	[5]	7	20
крок4	[0]	[7]	13	[5]	[7]	10
крок5	[0]	[7]	13	[5]	[7]	[10]
крок6	[0]	[7]	[13]	[5]	[7]	[10]

З таблиці видно, що шлях між вузлами 1 і 6 дорівнює 10.

Знайдемо матрицю найкоротших маршрутів між вузлами графа за алгоритмом Флойда:

$$\mathbf{F} = \begin{bmatrix} - & 7 & 13 & 5 & 7 & 10 \\ - & - & 6 & - & 8 & 11 \\ - & - & - & - & - & 6 \\ - & - & - & - & 2 & 5 \\ - & - & - & - & - & 3 \\ - & - & - & - & - & - \end{bmatrix}$$

В цій матриці  $(\mathbf{F})_{ij}$  дорівнює вазі шляху між вузлами  $i$  і  $j$ . Легко бачити, що за результатом алгоритму Флойда вага шляху з 1-го вузла до 6-го дорівнює 10.

## 5 Основи аналітичного аналізу дискретно- безперервних процесів за допомогою “Max-Plus” алгебри

Підхід до застосування апарата “Max-Plus” алгебри для опису дискретно-безперервних процесів можна пояснити на найпростішому прикладі транспортної системи [71].

Наприклад, дві залізничні станції  $S_1$  і  $S_2$  зв’язані між собою рейковими шляхами. Загальна схема транспортної системи представлена на рис. 5.1.

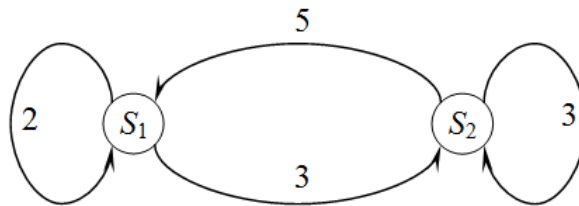


Рисунок 5.1 – Граф найпростішої залізничної системи

Ця система складається з внутрішнього і двох зовнішніх контурів. На зовнішніх і внутрішніх контурах є проміжні станції для посадки і висадки пасажирів, які не показані на рисунку 5.1, тому що вони не грають принципової ролі в розглянутій моделі. Припустимо, що є чотири поїзди, по два на кожній станції. В деякий момент часу вони одночасно відправляються, кожний своїм шляхом. Поїзди досягають іншу або ту ж саму станцію після певного часу, що відображено на рис. 5.1. Поїзди, що прибувають на станції  $S_1$  і  $S_2$  мають чекати один одного, щоб дозволити пасажирам зробити пересадку.

Припустимо, що немає ніякого розкладу і поїзди після того, як пасажири зроблять пересадку, відправляються знову. Час, необхідний для

пересадки, умовно включено до часу руху та інтерпретується як «час зайнятості» за аналогією з термінами мереж Петрі. Процес відправлення і прибуття поїздів можна розглядати як завгодно довго в розвитку. Якщо позначити вихідний час чергового відправлення через  $x_i(k+1)$  для  $(k+1)$ -го відправлення поїзда зі станції  $S_i$  ( $i=1,2$ ), то систему можна описати в такий спосіб:

$$\begin{aligned} x_1(k+1) &= \max[x_1(k) + 2, \quad x_2(k) + 5], \\ x_2(k+1) &= \max[x_1(k) + 3, \quad x_2(k) + 3], \end{aligned} \quad (5.1)$$

для  $k = 0, 1, 2, \dots$

Якщо прийняти, що  $x_1(0) = 0$  і  $x_2(0) = 0$ , то розвиток процесу в часі може бути представлено в такий спосіб:

$$\begin{array}{ccccccccc} \begin{bmatrix} 0 \\ 0 \end{bmatrix} & \rightarrow & \begin{bmatrix} 5 \\ 3 \end{bmatrix} & \rightarrow & \begin{bmatrix} 8 \\ 8 \end{bmatrix} & \rightarrow & \begin{bmatrix} 13 \\ 11 \end{bmatrix} & \rightarrow & \begin{bmatrix} 16 \\ 16 \end{bmatrix} & \rightarrow & \dots, \\ x_i(0) & & x_i(1) & & x_i(2) & & x_i(3) & & x_i(4) & & \end{array} \quad (5.2)$$

де:  $x_i(k)$  – вектор стану системи,

$x_i(0)$  – початкові умови для системи ( $i = 1, 2$ ).

Тобто, маємо періодично повторюваний розв’язок, період якого дорівнює 2 і середній час між відправленнями зі станцій дорівнює 4.

З точки зору теорії розкладу це істотний недолік. Бажано, щоб відправлення було більш регулярним, тобто щоб поїзди відправлялися через рівні проміжки часу.

Якщо змінити початкові умови на:

$$x_i(0) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad (5.3)$$

то розвиток процесу в часі згідно з рівняннями (5.1) матиме вигляд:

$$\begin{array}{ccccccccc} \begin{bmatrix} 1 \\ 0 \end{bmatrix} & \rightarrow & \begin{bmatrix} 5 \\ 4 \end{bmatrix} & \rightarrow & \begin{bmatrix} 9 \\ 8 \end{bmatrix} & \rightarrow & \begin{bmatrix} 13 \\ 12 \end{bmatrix} & \rightarrow & \begin{bmatrix} 17 \\ 16 \end{bmatrix} & \rightarrow & \dots \\ x_i(0) & & x_i(1) & & x_i(2) & & x_i(3) & & x_i(4) & & \end{array} \quad (5.4)$$

Період повторення став дорівнювати 1 і час між відправленнями поїздів зі станцій однаковий і дорівнює 4.

Задамося питанням, чи не можна для цих же умов зменшити час між відправленнями поїздів зі станцій?

Очевидно, що для подолання повного контуру між двома станціями  $S_1$  і  $S_2$  кожний поїзд витрачає  $3 + 5 = 8$  одиниць часу, тому середній час між відправленнями обмежений і він не може бути менше, ніж  $8/2 = 4$  одиниці. Для того, щоб зменшити час між сусідніми відправленнями, треба змінити саму транспортну систему. Умовно додамо ще один поїзд на внутрішній контур. Припустимо, що спочатку цей поїзд перебуває на станції  $S_1$ . Тоді рівняння розвитку процесу приймуть вигляд:

$$\begin{array}{l} x_1(k+1) = \max[x_1(k) + 2, \quad x_2(k) + 5], \\ x_2(k+1) = \max[x_1(k-1) + 3, \quad x_2(k) + 3], \end{array} \quad (5.5)$$

які в формі рівнянь (5.1) матимуть вигляд:

$$\begin{array}{l} x_1(k+1) = \max[x_1(k) + 2, \quad x_2(k) + 5], \\ x_2(k+1) = \max[x_3(k) + 3, \quad x_2(k) + 3], \\ x_3(k+1) = x_1(k). \end{array} \quad (5.6)$$

Для початкових умов  $x_1(0) = 0$ ,  $x_2(0) = 0$  і  $x_3(0) = 0$  розвиток процесу в часі для такої системи матиме вигляд:

$$\begin{array}{ccccccccc} \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} & \rightarrow & \begin{bmatrix} 5 \\ 3 \\ 0 \end{bmatrix} & \rightarrow & \begin{bmatrix} 8 \\ 6 \\ 5 \end{bmatrix} & \rightarrow & \begin{bmatrix} 11 \\ 9 \\ 8 \end{bmatrix} & \rightarrow & \begin{bmatrix} 14 \\ 12 \\ 11 \end{bmatrix} & \rightarrow & \dots, \\ x_i(0) & & x_i(1) & & x_i(2) & & x_i(3) & & x_i(4) & & \end{array} \quad (5.7)$$

Неважко помітити, що після деякого перехідного режиму, протягом двох тактів у системі встановлюється стійкий стан з періодом повторення, що дорівнює 1, і з часом між відправленнями поїздів зі станцій, що дорівнює 3. Значення часу між відправленнями обумовлено зовнішнім контуром на станції  $S_2$ . На цьому контурі курсує один поїзд, для якого потрібно 3 одиниці часу, щоб його проїхати. Внутрішній контур привів би до більш низького значення часу послідовного відправлення поїздів  $8/3$  одиниць (час контуру, ділений на число поїздів). Якщо умовно додати поїзд на зовнішній контур станції  $S_2$ , то рівняння (5.6) приймають вигляд:

$$\begin{aligned} x_1(k+1) &= \max[x_1(k) + 2, \quad x_2(k) + 5], \\ x_2(k+1) &= \max[x_3(k) + 3, \quad x_2(k-1) + 3], \\ x_3(k+1) &= x_1(k). \end{aligned} \quad (5.8)$$

Використовуючи рівняння (5.1), рівняння (5.8) матимуть такий вигляд:

$$\begin{aligned} x_1(k+1) &= \max[x_1(k) + 2, \quad x_2(k) + 5], \\ x_2(k+1) &= \max[x_3(k) + 3, \quad x_4(k) + 3], \\ x_3(k+1) &= x_1(k), \\ x_4(k+1) &= x_2(k). \end{aligned} \quad (5.9)$$

При нульових початкових умовах поведінка транспортної системи описується такою послідовністю векторів:

$$\begin{aligned} \dots \rightarrow \begin{bmatrix} 10 \\ 8 \\ 8 \\ 3 \end{bmatrix} &\rightarrow \begin{bmatrix} 13 \\ 11 \\ 10 \\ 8 \end{bmatrix} \rightarrow \begin{bmatrix} 16 \\ 13 \\ 13 \\ 11 \end{bmatrix} \rightarrow \begin{bmatrix} 18 \\ 16 \\ 16 \\ 13 \end{bmatrix} \rightarrow \begin{bmatrix} 21 \\ 19 \\ 18 \\ 16 \end{bmatrix} \rightarrow \dots \\ x_i(3) & \quad x_i(4) \quad x_i(5) \quad x_i(6) \quad x_i(7) \end{aligned} \quad (5.10)$$

Така система характеризується періодом повторення станів, що дорівнює 3, і середнім часом між відправленням поїздів зі станцій, який

дорівнює  $8/3$ , що обумовлено вже внутрішнім контуром. Якщо взяти початкові умови  $x_1(0) = 5$ ,  $x_2(0) = 8/3$ ,  $x_3(0) = 7/3$  і  $x_4(0) = 0$ , то одержимо період повторення станів, рівний 1, з тим же середнім часом між відправленням поїздів зі станцій, рівним  $8/3$ .

Канонічна форма розглянутої системи в загальному виді може бути представлена в такому вигляді:

$$\begin{aligned} x_i(k+1) &= \max[a_{i1} + x_1(k), a_{i2} + x_2(k), \dots, a_{in} + x_n(k)] = \\ &= \max_j [a_{ij} + x_j(k)], \quad i = 1, \dots, n. \end{aligned} \quad (5.11)$$

Використовуючи позначення “Max-Plus” алгебри, маємо:

$$x_i(k+1) = \oplus_j (a_{ij} \otimes x_j(k)), \quad i = 1, \dots, n, \quad (5.12)$$

яке у векторному вигляді буде записано:

$$\mathbf{x}(k+1) = \mathbf{A} \otimes \mathbf{x}(k). \quad (5.13)$$

Очевидно, що лінійне різницеве представлення системи в формі (5.13) аналогічно представленню лінійних дискретних систем за допомогою звичайної алгебри в часовій області простору змінних стану. Однак, незважаючи на спільність представлення за формою, в дискретно-безперервних системах в це рівняння вкладається інший зміст.

Якщо задатися початковими умовами для  $\mathbf{x}(0) = \mathbf{x}_0$ , то:

$$\begin{aligned} \mathbf{x}(1) &= \mathbf{A} \otimes \mathbf{x}_0, \\ \mathbf{x}(2) &= \mathbf{A} \otimes \mathbf{x}(1) = \mathbf{A} \otimes (\mathbf{A} \otimes \mathbf{x}_0) = (\mathbf{A} \otimes \mathbf{A}) \otimes \mathbf{x}_0 = \mathbf{A}^2 \otimes \mathbf{x}_0. \end{aligned} \quad (5.14)$$

Можна показати, що дійсно  $\mathbf{A} \otimes (\mathbf{A} \otimes \mathbf{x}_0) = (\mathbf{A} \otimes \mathbf{A}) \otimes \mathbf{x}_0$ . Для простоти замість  $\mathbf{A} \otimes \mathbf{A}$  ми просто запишемо  $\mathbf{A}^2$ . В загальному випадку:

$$\mathbf{x}(k) = \underbrace{(\mathbf{A} \otimes \mathbf{A} \otimes \dots \otimes \mathbf{A})}_{k \text{ разів}} \otimes \mathbf{x}_0 = \mathbf{A}^k \otimes \mathbf{x}_0. \quad (5.15)$$

Матриці  $\mathbf{A}^2, \mathbf{A}^3, \dots$  можуть бути безпосередньо обчислені. Розглянемо матрицю  $\mathbf{A}$  з (5.1):

$$\mathbf{A}^2 = \begin{pmatrix} \max[2+2, 5+3] & \max[2+5, 5+3] \\ \max[3+2, 3+3] & \max[3+5, 3+3] \end{pmatrix} = \begin{pmatrix} 8 & 8 \\ 6 & 8 \end{pmatrix},$$

в загальному випадку:

$$(\mathbf{A}^2)_{ij} = \bigoplus_l a_{il} \otimes a_{lj} = \max_l [a_{il} + a_{lj}]. \quad (5.16)$$

У термінах прикладу залізничної мережі  $(\mathbf{A}^2)_{ij}$  може інтерпретуватися як максимум (відносно) всіх зв'язків від станції  $S_j$  через станцію  $S_l$  до станції  $S_i$ . Мова іде про зв'язки детермінованої довжини між станціями  $S_j$  і  $S_i$ . У термінології теорії графів, станції називають вузлами, а зв'язки між станціями називають дугами. В загальному випадку,  $(\mathbf{A}^k)_{ij}$  позначає максимум всіх зв'язків довжини  $k$ , що починаються у вузлі  $j$  і закінчуються у вузлі  $i$ .

Розглянемо деяку дискретно-безперервну систему з входами і виходами, що описується наступною канонічною формою:

$$\begin{aligned} x_i(k+1) &= \max \begin{bmatrix} a_{i1} + x_1(k), \dots, a_{in} + x_n(k), \\ b_{i1} + u_1(k), \dots, b_{im} + u_m(k) \end{bmatrix}, \quad i = 1, \dots, n; \\ y_i(k) &= \max [c_{i1} + x_1(k), \dots, c_{in} + x_n(k)], \quad i = 1, \dots, p. \end{aligned} \quad (5.17)$$

Векторно-матричне представлення якої має вигляд:

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{Ax}(k) \oplus \mathbf{Bu}(k), \\ \mathbf{y}(k) &= \mathbf{Cx}(k). \end{aligned} \quad (5.18)$$

В цьому випадку перше рівняння в (5.18) є неявним відносно  $\mathbf{x}(k+1)$ :

$$\mathbf{x}(k+1) = \mathbf{A}_0 \mathbf{x}(k+1) \oplus \mathbf{A}_1 \mathbf{x}(k) \oplus \dots \oplus \mathbf{A}_{l+1} \mathbf{x}(k-l). \quad (5.19)$$



Виключаючи  $\mathbf{x}(k+1)$  з правої частини рівняння (5.19), маємо наступне рівняння:

$$\mathbf{x}(k+1) = \mathbf{A}_0^* \mathbf{A}_1 \mathbf{x}(k) \oplus \dots \oplus \mathbf{A}_0^* \mathbf{A}_{l+1} \mathbf{x}(k-l), \quad (5.20)$$

де

$$\mathbf{A}_0^* = \mathbf{I} \oplus \mathbf{A}_0 \oplus \mathbf{A}_0^2 \oplus \mathbf{A}_0^3 \oplus \dots$$

Матриця  $\mathbf{I}$  називається в “Max-Plus” алгебрі матрицею ідентичності: вона має нулі на головній діагоналі, а всі інші елементи  $\varepsilon$ . Рівняння (5.20) має сенс, тільки якщо  $\mathbf{A}_0^*$  повністю визначена (її елементи скінчені або дорівнюють  $\varepsilon$ ). Це має місце, якщо пріоритетний граф для  $\mathbf{A}$  не містить замкнуті контури, оскільки тоді  $\mathbf{A}^k = \varepsilon$  для  $k \geq n$ . В цьому випадку рівняння (5.20) може бути переписане як перше рівняння системи (5.18). Це стандартний підхід в теорії керування, який часто використовується в класичних лінійних безперервних і дискретних динамічних системах.

## 6 Приклади аналітичного опису процесів в дискретно-безперервних виробничих системах

### 6.1 Виробнича система з паралельно-послідовною обробкою виробів

Розглянемо найпростішу виробничу систему, зображену на рис. 6.1. Виробнича система складається з чотирьох виробничих ділянок:  $M_1$ ,  $M_2$ ,  $M_3$ ,  $M_4$ . Сировина подається в  $M_1$ ,  $M_2$  і  $M_3$ , де відбувається обробка і потім відсилається в  $M_4$ , де відбувається збирання. Час транспортування від ділянки до ділянки і час обробки на ділянці показані на рисунку:

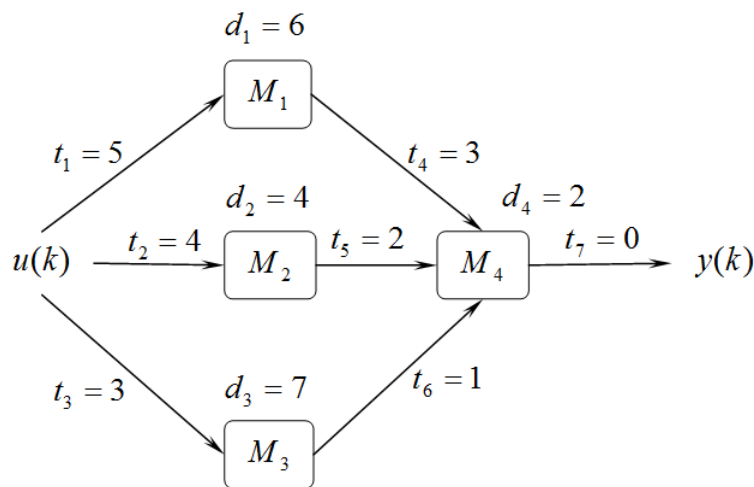


Рисунок 6.1 – Виробнича система з паралельно-послідовною обробкою виробів

Визначимо, що:

- $u(k+1)$  – момент часу, в який сировина подається в систему в  $(k+1)$ -й раз;
- $x_i(k)$  – момент часу, в який  $k$ -й раз починає роботу  $i$ -а виробнича ділянка;

- $y(k)$  – момент часу, в який  $k$ -й (кінцевий) продукт залишає систему.

Тепер визначимо момент часу, в який ділянка  $M_1$  працює в  $(k+1)$ -й раз. Якщо сировина подається в систему  $(k+1)$ -й раз, то вона стає доступною на вході ділянки  $M_1$  в момент часу  $t = u(k+1) + t_1 = u(k+1) + 5$ . При цьому  $M_1$  може починати нову операцію тільки по закінченні попередньої, тобто  $k$ -ї. З початку обробки на  $M_1$  час обробки  $d_1$  становить 6 одиниць часу,  $k$ -й проміжний продукт залишить  $M_1$  в момент часу  $t = x_1(k) + 6$ . Після того, як в  $M_1$  починається обробка нового об'єму сировини, а попередній продукт вже залишив виробничу ділянку  $M_1$ , маємо:

$$\forall k \in N_0 : x_1(k+1) = \max(x_1(k) + 6, u(k+1) + 5). \quad (6.1)$$

Умова, що спочатку виробнича ділянка  $M_1$  порожня і існує резерв сировини, відповідає початковій умові  $x_1(0) = \varepsilon$ . Тоді, як витікає з рівняння (6.1),  $x_1(1) = u(1) + 5$ , тобто перша партія сировини посилається в систему і буде негайно оброблена (після затримки на 5 часових інтервалів, необхідних для транспортування сировини від входу системи до  $M_1$ ).

Шляхом подібних міркувань знайдемо вирази для моментів часу, в які  $M_2$ ,  $M_3$  і  $M_4$  починають  $(k+1)$ -й цикл роботи і момент часу, в який  $k$ -й кінцевий продукт залишає систему:

$$\begin{aligned} \forall k \in N_0 : \\ x_2(k+1) &= \max(x_2(k) + 4, u(k+1) + 4), \\ x_3(k+1) &= \max(x_3(k) + 7, u(k+1) + 3), \\ x_4(k+1) &= \max(x_1(k+1) + 6 + 3, x_2(k+1) + 4 + 2, x_3(k+1) + 7 + 1, x_4(k) + 2) = \\ &= \max(x_1(k) + 15, x_2(k) + 10, x_3(k) + 15, x_4(k) + 2, u(k+1) + 14), \\ y(k) &= x_4(k) + 2 + 0. \end{aligned} \quad (6.2)$$

Умова, що спочатку всі бункери порожні відповідає такій початковій умові:  $x_1(0) = x_2(0) = x_3(0) = x_4(0) = \varepsilon$ .

Використовуючи позначення “Max-plus”, система рівнянь (6.1), (6.2)

матиме вигляд:

$$\begin{aligned} x_1(k+1) &= 6x_1(k) \oplus 5u(k+1), \\ x_2(k+1) &= 4x_2(k) \oplus 4u(k+1), \\ x_3(k+1) &= 7x_3(k) \oplus 3u(k+1), \\ x_4(k+1) &= 15x_1(k) \oplus 10x_2(k) \oplus 15x_3(k) \oplus 2x_4(k) \oplus 14u(k+1), \\ y(k) &= 2x_4(k), \end{aligned} \quad (6.3)$$

або в матричному вигляді:

$$\mathbf{x}(k+1) = \mathbf{Ax}(k) + \mathbf{Bu}(k+1) = \begin{bmatrix} 6 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & 4 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & 7 & \varepsilon \\ 15 & 10 & 15 & 2 \end{bmatrix} \mathbf{x}(k) \oplus \begin{bmatrix} 5 \\ 4 \\ 3 \\ 14 \end{bmatrix} \mathbf{u}(k+1) \quad (6.4)$$

$$\mathbf{y}(k) = \mathbf{Cx}(k) = [\varepsilon \quad \varepsilon \quad \varepsilon \quad 2] \mathbf{x}(k), \quad (6.5)$$

де  $x(k) = \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k) \end{bmatrix}$ .

Результати моделювання (Додаток, лістинг 1) для системи (6.4) при  $\mathbf{u}(k) = 0 = const$  (X – послідовність векторів стану, починаючи з  $x(1)$ , Y – вихідні значення вектора виходу):

X =

5	11	17	23	29	35	41	47	53	59
4	8	12	16	20	24	28	32	36	40
3	10	17	24	31	38	45	52	59	66
14	20	26	32	39	46	53	60	67	74

Y =

16	22	28	34	41	48	55	62	69	76
----	----	----	----	----	----	----	----	----	----

Отримані результати можна графічно представити у вигляді діаграми Ганта (рис. 6.2). На осі ординат наведені номери виробничих ділянок, а на осі абсцис – моменти часу обробки та транспортування виробів.

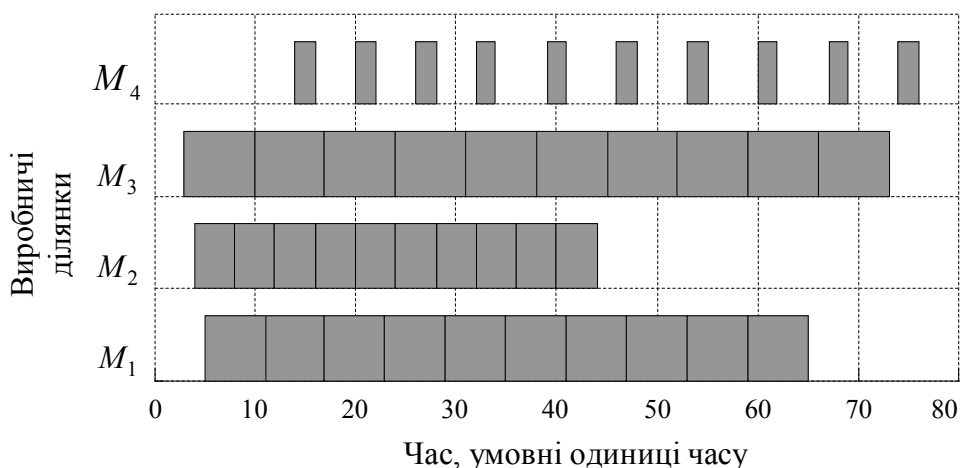


Рисунок 6.2 – Діаграма Ганта для виробничої системи з рис. 6.1

## 6.2 Виробнича система зі змішаною послідовністю обробки виробів

Розглянемо просту виробничу систему, зображену на рис. 6.3. Ця виробнича система складається з чотирьох виробничих ділянок:  $M_1$ ,  $M_2$ ,  $M_3$ ,  $M_4$ . Сировина подається в  $M_1$  і  $M_2$ , де відбувається обробка, потім продукт з ділянки  $M_1$  і частина продукту з  $M_2$  посилається до  $M_3$ . Збирання здійснюється в ділянці  $M_4$ , куди надходить оброблений продукт з  $M_3$  і та частина, що залишилася, з  $M_2$ . Час обробки для  $M_1$ ,  $M_2$ ,  $M_3$ ,  $M_4$  відповідно дорівнює  $d_1 = 6$ ,  $d_2 = 4$ ,  $d_3 = 2$ ,  $d_4 = 7$  одиниць часу. Час транспортування сировини до ділянок  $M_1$  і  $M_2$  дорівнює, відповідно,  $t_1 = 5$ ,  $t_2 = 4$  одиниць часу, а транспортування від  $M_1$  і  $M_2$  до  $M_3$  –  $t_3 = 3$  і  $t_4 = 3$  одиниць часу відповідно. З ділянки  $M_2$  до  $M_4$  продукт надходить за  $t_5 = 2$ , від  $M_3$  до  $M_4$  за  $t_6 = 1$  одиниць часу. Транспортування готового

продукту вимагає  $t_7 = 1$  одиниць часу. На вході системи і між виробничими ділянками існують бункери, ємність яких достатня для того, щоб не відбувалося їх переповнення. Припустимо, що спочатку бункери порожні і жодна з виробничих ділянок не містить сировини або проміжних продуктів. Проміжна ділянка починає роботу над новим виробом лише по закінченні роботи над попереднім. Будемо вважати, що кожна виробнича ділянка починає роботу, як тільки будуть готові всі її компоненти.

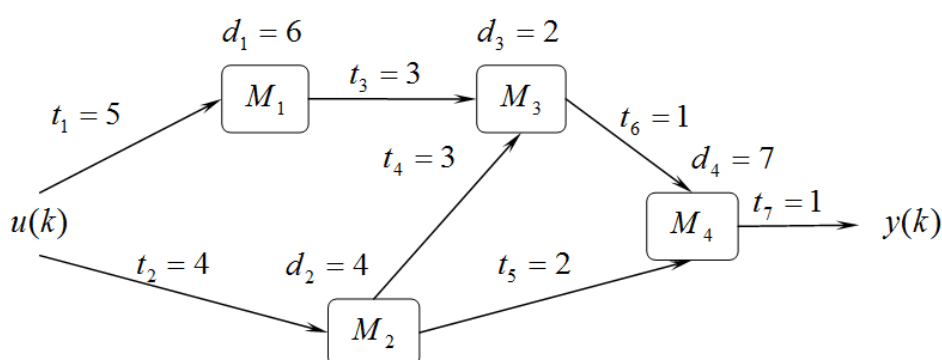


Рисунок 6.3 – Виробнича система зі змішаною обробкою виробів

Припустимо  $k \in \mathbf{N}_0$ . Позначимо, що

- $u(k+1)$  – момент часу, в який сировина подається в систему в  $(k+1)$ -й раз;
- $x_i(k)$  – момент часу, в який  $k$ -й раз починає роботу  $i$ -а ( $i=1,2,3,4$ ) виробнича ділянка;
- $y(k)$  – момент часу, в який  $k$ -й кінцевий продукт залишає систему.

Визначимо момент часу, коли ділянка  $M_1$  працює в  $(k+1)$ -й раз. Відомо, якщо сировина подається в систему в  $(k+1)$ -й раз, то вона стане доступною на вході  $M_1$  в момент часу  $t = u(k+1) + t_1$ . При цьому,  $M_1$  може починати нову операцію лише по закінченні попередньої, тобто  $k$ -ї. З початку обробки виробів на  $M_1$  час обробки становить  $d_1 = 6$  одиниць

часу. Тоді  $k$ -й проміжний продукт залишить  $M_1$  у момент часу  $t = x_1(k) + d_1$ . Таким чином, після початку обробки сировини в  $M_1$  та по мірі її готовності і коли поточний продукт вже залишив виробничу ділянку  $M_1$ , маємо наступне рівняння:

$$\forall k \in N_0 : x_1(k+1) = \max(x_1(k) + d_1, u(k+1) + t_1) \quad (6.6)$$

Початкова умова, що виробнича ділянка  $M_1$  порожня і існує резерв сировини має вигляд:  $x_1(0) = \varepsilon$ . Тоді з рівняння (6.6) витікає:  $x_1(1) = u(1) + t_1$ . Тобто перша партія сировини посилається в систему і буде оброблена з затримкою на 5 інтервалів часу, які необхідні для транспортування сировини від входу системи до  $M_1$  (рис. 6.3).

Шляхом подібних міркувань знайдемо вирази для моментів часу, в які  $M_2$ ,  $M_3$ ,  $M_4$  почнуть  $(k+1)$ -й цикл роботи та момент часу, коли  $k$ -й кінцевий продукт залишить систему:

$$\forall k \in N_0 :$$

$$x_2(k+1) = \max(x_2(k) + d_2, u(k+1) + t_2),$$

$$\begin{aligned} x_3(k+1) &= \max(x_1(k+1) + d_1 + t_3, x_2(k+1) + d_2 + t_4, x_3(k) + d_3) = \\ &= \max(x_1(k) + d_1 + d_1 + t_3, x_2(k) + d_2 + d_2 + t_4, x_3(k) + d_3, \\ &\quad u(k+1) + t_1 + d_1 + t_3, u(k+1) + t_2 + d_2 + t_4), \end{aligned}$$

$$\begin{aligned} x_4(k+1) &= \max(x_2(k+1) + d_2 + t_5, x_3(k+1) + d_3 + t_6, x_4(k) + d_4) = \\ &= \max(x_1(k) + d_1 + d_1 + t_3 + d_3 + t_6, \\ &\quad x_2(k) + d_2 + d_2 + t_4 + d_3 + t_6, x_2(k) + d_2 + d_2 + t_5, \\ &\quad x_3(k) + d_3 + d_3 + t_6, x_4(k) + d_4, \\ &\quad u(k+1) + t_1 + d_1 + t_3 + d_3 + t_6, u(k+1) + t_2 + d_2 + t_4 + d_3 + t_6, \\ &\quad u(k+1) + t_2 + d_2 + t_5), \end{aligned}$$

$$y(k) = x_4(k) + d_4 + t_7.$$

Підставимо в вищенаведену систему числові значення параметрів  $d_i$  ( $i = \overline{1,4}$ ) і  $t_j$  ( $j = \overline{1,7}$ ). Одержимо:

$$\forall k \in N_0 :$$

$$x_1(k+1) = \max(x_1(k) + 6, u(k+1) + 5),$$

$$x_2(k+1) = \max(x_2(k) + 4, u(k+1) + 4),$$

$$\begin{aligned} x_3(k+1) &= \max(x_1(k) + 15, x_2(k) + 11, x_3(k) + 2, u(k+1) + 14, u(k+1) + 11) = \\ &= \max(x_1(k) + 15, x_2(k) + 11, x_3(k) + 2, u(k+1) + 14), \end{aligned}$$

$$\begin{aligned} x_4(k+1) &= \max(x_1(k) + 18, x_2(k) + 14, x_2(k) + 10, x_3(k) + 5, x_4(k) + 7, \\ &\quad u(k+1) + 17, u(k+1) + 14, u(k+1) + 10) = \\ &= \max(x_1(k) + 18, x_2(k) + 14, x_3(k) + 5, x_4(k) + 7, u(k+1) + 17), \end{aligned}$$

$$y(k) = x_4(k) + 8.$$

Початкова умова, що спочатку всі бункери порожні, має вигляд:

$$x_1(0) = x_2(0) = x_3(0) = x_4(0) = \varepsilon.$$

В термінах “Мах-plus” алгебри:

$$x_1(k+1) = 6x_1(k) \oplus 5u(k+1),$$

$$x_2(k+1) = 4x_2(k) \oplus 4u(k+1),$$

$$x_3(k+1) = 15x_1(k) \oplus 11x_2(k) \oplus 2x_3(k) \oplus 14u(k+1), \quad (6.7)$$

$$x_4(k+1) = 18x_1(k) \oplus 14x_2(k) \oplus 5x_3(k) \oplus 7x_4(k) \oplus 17u(k+1),$$

$$y(k) = 8x_4(k).$$

Запишемо (6.7) в матричному вигляді:

$$\mathbf{x}(k+1) = \mathbf{Ax}(k) + \mathbf{Bu}(k+1) = \begin{bmatrix} 6 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & 4 & \varepsilon & \varepsilon \\ 15 & 11 & 2 & \varepsilon \\ 18 & 14 & 5 & 7 \end{bmatrix} \mathbf{x}(k) \oplus \begin{bmatrix} 5 \\ 4 \\ 14 \\ 17 \end{bmatrix} \mathbf{u}(k+1), \quad (6.8)$$

$$\mathbf{y}(k) = \mathbf{Cx}(k) = [\varepsilon \quad \varepsilon \quad \varepsilon \quad 8] \mathbf{x}(k),$$



$$\text{де } \mathbf{x}(k) = \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k) \end{bmatrix}.$$

Результати моделювання (Додаток, лістинг 2) для системи (6.8) при  $\mathbf{u}(k) = 0 = \text{const}$  ( $X$  – послідовність векторів стану, починаючи з  $x(1)$ ,  $Y$  – вихід системи) приведені нижче:

$X =$

5	11	17	23	29	35	41	47	53	59
4	8	12	16	20	24	28	32	36	40
14	20	26	32	38	44	50	56	62	68
17	24	31	38	45	52	59	66	73	80

$Y =$

25	32	39	46	53	60	67	74	81	88
----	----	----	----	----	----	----	----	----	----

Графічне представлення результатів у вигляді діаграми Ганта наведено на рис. 6.4.

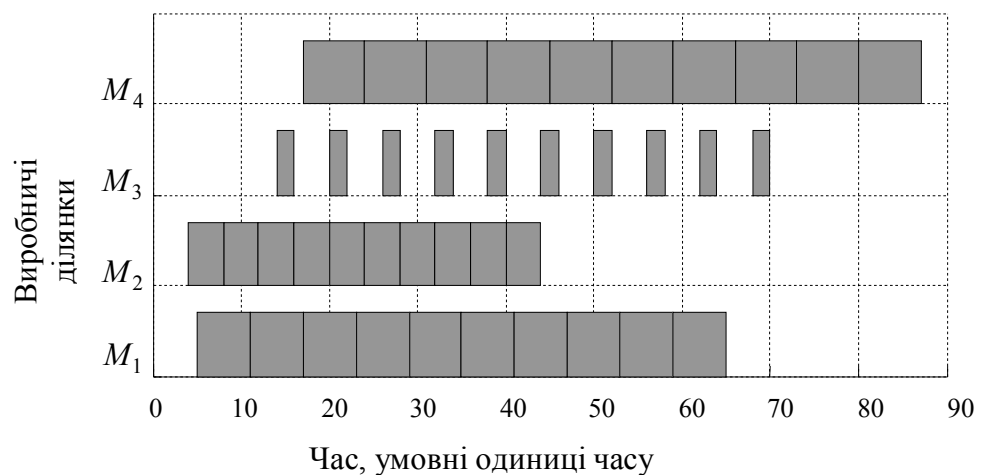


Рисунок 6.4 – Діаграма Ганта для виробничої системи з рис. 6.3

### 6.3 Багаторівнева виробнича система

Розглянемо виробничу систему, що складається з чотирьох виробничих ділянок  $M_1$ ,  $M_2$ ,  $M_3$  і  $M_4$ . В виробничій системі виробляються чотири різних типи виробів ( $P_1$ ,  $P_2$ ,  $P_3$  і  $P_4$ ) відповідно до певного виробничого циклу. Маршрути, якими ідуть різні типи виробів, наведені на рис. 6.5.

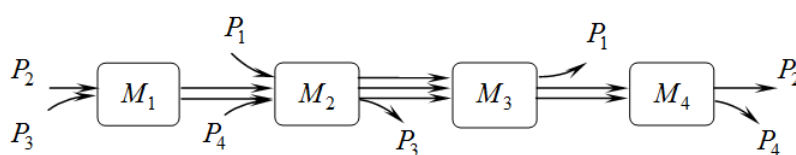


Рисунок 6.5 – Схема обробки різних типів виробів на виробничих ділянках

Вироби типу  $P_1$  спочатку приходять на ділянку  $M_2$ , а далі – на  $M_3$ . Вироби типу  $P_2$  попадають в систему через ділянку  $M_1$ , далі направляються до ділянки  $M_2$ , потім до ділянки  $M_3$  та залишають систему з ділянки  $M_4$ . Вироби типу  $P_3$  спочатку обробляються на ділянці  $M_1$ , далі –  $M_2$ . Вироби типу  $P_4$  попадають в систему на ділянку  $M_2$ , далі –  $M_3$ ,  $M_4$ . Припустимо, що:

- Вироби переміщуються на платформах. Використовується лише одна платформа для кожного типу виробів.
- Час транспортування малий і він не враховується.
- Відома послідовність обробки виробів на ділянках: на ділянці  $M_1$  це ( $P_2, P_3$ ), тобто ділянка спочатку обробляє виріб типу  $P_2$ , а потім – типу  $P_3$ , на ділянці  $M_2$  – послідовність обробки така:  $P_1, P_2, P_3, P_4$ , на ділянці  $M_3$  –  $P_1, P_2, P_4$ , на ділянці  $M_4$  –  $P_2, P_4$ . Будемо називати ці послідовності

правилами локального розподілу і позначати як  $\sigma$  (тобто  $\sigma_1$  для послідовності на  $M_1$ ,  $\sigma_2$  для послідовності на  $M_2$ ,  $\sigma_3$  для  $M_3$  та  $\sigma_4$  для  $M_4$ ).

Інформацію про послідовність операцій та час здійснення різних процесів (час обробки) показано на рис. 6.6. На даному рисунку операції представлено у вигляді впорядкованих пар  $(P_i, M_j)$  ( $i = \overline{1,4}$ ,  $j = \overline{1,4}$ ), які означають, що виріб  $P_i$  обробляється на ділянці  $M_j$ . Дуги – це пріоритетні переходи між операціями. Ліворуч зверху кожної операції позначається її тривалість. Наприклад,  $(P_1, M_2)$  має тривалість  $d_3 = 3$ .

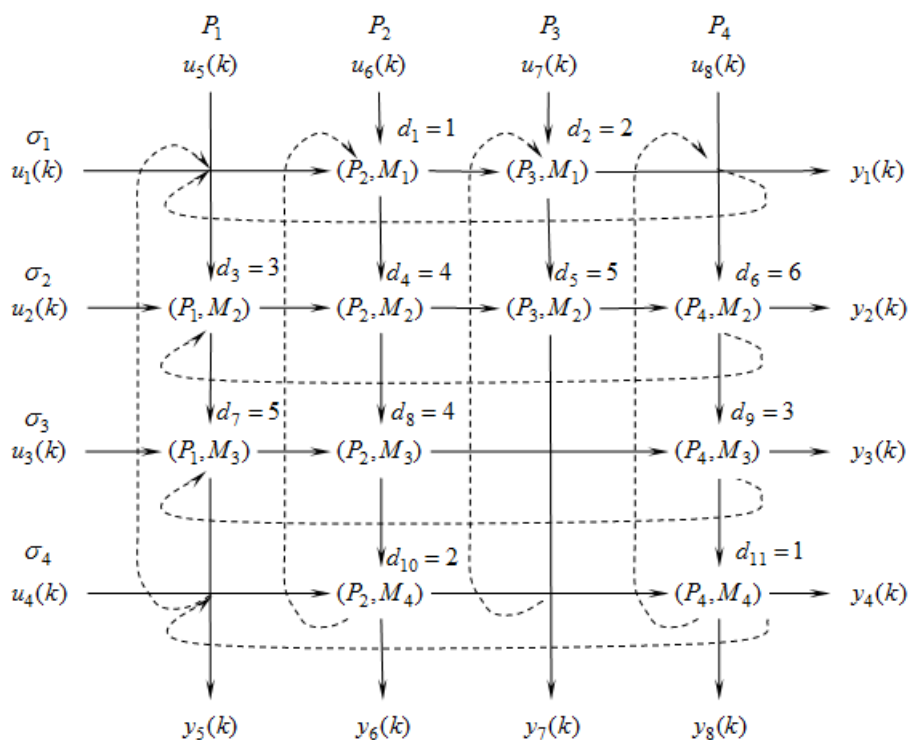


Рисунок 6.6 – Послідовність і тривалість різних операцій

Для спрощення процесу виведення рівнянь даної системи розглянемо, що відбувається в одному циклі виробничого процесу.

Введемо позначення:

- $u_i(k)$  – момент часу, в який ділянка  $M_i$  готова до першої дії і почне її в  $k$ -му виробничому циклі при  $i = 1, 2, 3, 4$ ;
- $u_j(k)$  – момент часу, в який заготівка для виробу типу  $P_{j-4}$  доступна в  $k$ -му виробничому циклі при  $j = 5, 6, 7, 8$ ;
- $x_i(k)$  – момент часу, у який починається  $i$ -а операція в  $k$ -у виробничому циклі для  $i = 1, 2, \dots, 11$ ;
- $y_i(k)$  – момент часу, у який на ділянці  $M_i$  завершиться обробка останнього виробу з  $k$ -о виробничого циклу, який повинен оброблятися на ділянці  $M_i$  при  $i = 1, 2, 3, 4$ ;
- $y_j(k)$  – момент часу, в який обробку кінцевого продукту типу  $P_{j-4}$   $k$ -го виробничого циклу буде завершено при  $j = 5, 6, 7, 8$ .

Визначимо момент часу, коли починається перша операція: обробка продукту  $P_2$  на ділянці  $M_1$  в  $(k + 1)$ -у виробничому циклі. Це відбудеться за виконання умов:

1) Ділянка  $M_1$  готова до нового циклу роботи. Для  $(k + 1)$ -го циклу це момент часу  $u_1(k + 1)$ .

2) Заготівка для виробу  $P_2$  доступна в  $(k + 1)$ -у циклі. Момент часу визначимо як  $u_6(k + 1)$ .

3) Ділянка  $M_1$  завершила обробку останнього виробу  $k$ -го циклу у момент часу, коли продукт  $P_3$  буде оброблено, відповідно до правила  $\sigma_1$ . Обробка продукту  $P_3$  на ділянці  $M_1$  в  $k$ -му виробничому циклі – це операція  $x_2(k)$ , її тривалість  $d_2 = 2$  одиниць часу.

4) Обробка продукту  $P_2$  завершена в  $k$ -му циклі. Обробка кінцевого продукту  $P_2$  в  $k$ -му виробничому циклі буде завершено операцією  $x_{10}(k)$  тривалістю  $d_{10} = 2$  одиниць часу.

Тоді рівняння в термінах “Max-plus” алгебри для  $x_1(k+1)$ :

$$x_1(k+1) = d_2x_2(k) \oplus d_{10}x_{10}(k) \oplus u_1(k+1) \oplus u_6(k+1). \quad (6.9)$$

Аналогічно для операцій  $x_i(k+1)$  ( $i = 2, 3, \dots, 11$ ) одержимо таку послідовність рівнянь:

$$\begin{aligned} x_2(k+1) &= d_1x_1(k+1) \oplus d_5x_5(k) \oplus u_7(k+1), \\ x_3(k+1) &= d_6x_6(k) \oplus d_7x_7(k) \oplus u_2(k+1) \oplus u_5(k+1), \\ x_4(k+1) &= d_1x_1(k+1) \oplus d_3x_3(k+1), \\ x_5(k+1) &= d_2x_2(k+1) \oplus d_4x_4(k+1), \\ x_6(k+1) &= d_5x_5(k+1) \oplus d_{11}x_{11}(k) \oplus u_8(k+1), \\ x_7(k+1) &= d_3x_3(k+1) \oplus d_9x_9(k) \oplus u_3(k+1), \\ x_8(k+1) &= d_4x_4(k+1) \oplus d_7x_7(k+1), \\ x_9(k+1) &= d_6x_6(k+1) \oplus d_8x_8(k+1), \\ x_{10}(k+1) &= d_8x_8(k+1) \oplus d_{11}x_{11}(k) \oplus u_4(k+1), \\ x_{11}(k+1) &= d_9x_9(k+1) \oplus d_{10}x_{10}(k+1). \end{aligned} \quad (6.10)$$

або:

$$\mathbf{x}(k+1) = \mathbf{A}_0\mathbf{x}(k+1) \oplus \mathbf{A}_1\mathbf{x}(k) \oplus \mathbf{B}_0\mathbf{u}(k+1) = \mathbf{A}\mathbf{x}(k) \oplus \mathbf{B}\mathbf{u}(k+1), \quad (6.11)$$

де  $\mathbf{A} = \mathbf{A}_0^* \otimes \mathbf{A}_1$  і  $\mathbf{B} = \mathbf{A}_0^* \otimes \mathbf{B}_0$ .

Аналіз розглянутої системи проведено з використанням Max-Plus Algebra Toolbox (Додаток, лістинг 3). Будемо вважати, що  $\mathbf{x}(0) = [\varepsilon]^{11 \times 1}$ , тобто всі ділянки не зайнято перед початком операції і  $u(1) = [e]^{8 \times 1}$ , тобто всі ділянки можуть бути запущені без затримки. Час циклу і послідовні стани  $\mathbf{x}(k)$  системи для даного прикладу наведено нижче.

$\mathbf{X} =$

0	14	32	50	68	86	104	122	140	158
1	15	33	51	69	87	105	123	141	159
0	18	36	54	72	90	108	126	144	162

3	21	39	57	75	93	111	129	147	165
7	25	43	61	79	97	115	133	151	169
12	30	48	66	84	102	120	138	156	174
3	21	39	57	75	93	111	129	147	165
8	26	44	62	80	98	116	134	152	170
18	36	54	72	90	108	126	144	162	180
12	30	48	66	84	102	120	138	156	174
21	39	57	75	93	111	129	147	165	183

Y =

3	17	35	53	71	89	107	125	143	161
18	36	54	72	90	108	126	144	162	180
21	39	57	75	93	111	129	147	165	183
22	40	58	76	94	112	130	148	166	184
8	26	44	62	80	98	116	134	152	170
14	32	50	68	86	104	122	140	158	176
12	30	48	66	84	102	120	138	156	174
22	40	58	76	94	112	130	148	166	184

Зайнятість ділянок в часі наведено на діаграмі Ганта (рис. 6.7).

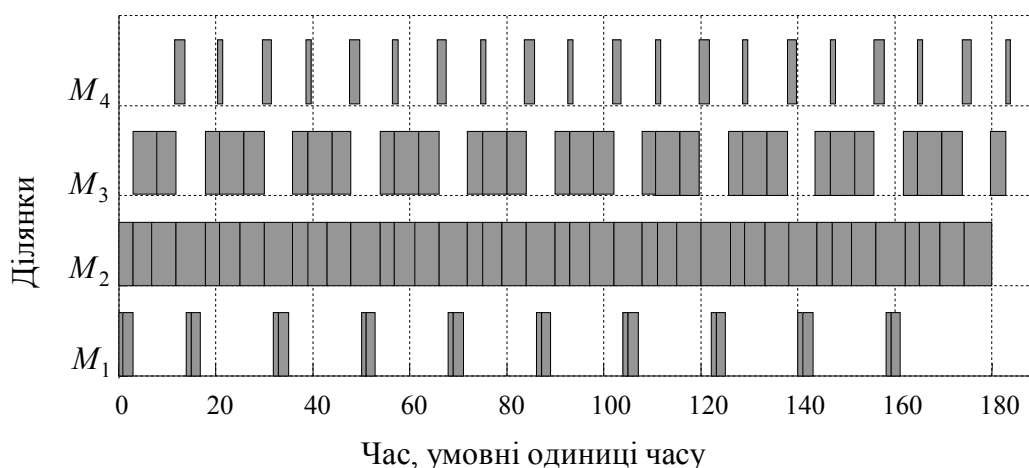


Рисунок 6.7 – Діаграма Ганта для багаторівневої виробничої системи

#### 6.4 Телекомунікаційна мережа з пакетною комутацією

Загальний підхід до моделювання процесів в телекомунікаційній мережі полягає в наступному. Припустимо мережа складається з  $n$  вузлів, в кожному з яких є одноканальний пристрій обробки заявок (пакетів) і буфер, в якому може відбуватися очікування обробки в черзі. Топологія мережі задається орієнтованим ациклічним графом  $G(V, \epsilon)$ , де  $V = \{1, \dots, n\}$  – множина вершин графа, які відповідають вузлам мережі, а  $\epsilon = \{(i, j)\} \in V \times V$  – множина дуг графа, що характеризують маршрути передачі заявок мережею.

В структурі мережі виділяють особливі вузли. Множина вузлів мережі  $P(i)$ , які називаються джерелами і формують нескінченний потік заявок (пакетів), що надходять до мережі. Вузли, через які заявки покидають мережу після обслуговування в них, позначають через  $S(i)$  і називаються множиною вузлів-приймачів заявок.

В початковий момент часу всі пристрої обробки заявок в мережі вільні, черга заявок в кожному вузлі-джерелі має нескінченну довжину, а черги всіх інших вузлів  $i$  містять по  $r_i$  заявок, які готові до обслуговування пристроями.

Для формалізації процесу вводяться наступні позначення:

$\tau_{ik}$  – тривалість обслуговування  $k$ -ї заявки в  $i$ -му вузлі мережі;

$x_i(k)$  – момент часу завершення обслуговування  $k$ -ї заявки в  $i$ -у вузлі мережі;

$a_i(k)$  – момент надходження  $k$ -ї заявки в чергу на обслуговування в  $i$ -у вузлі мережі.

Припустимо, що  $\tau_{ik}$  – невід’ємні випадкові величини з математичним очікуванням  $E[\tau_{ik}] < \infty$  для всіх  $i = 1, \dots, n$  та  $k = 1, 2, \dots$

Для визначеності задаються початкові умови процесу:

$$x_i(0) = 0, \quad x_i(k) = -\infty \text{ для } k < 0 \text{ і } i = 1, 2, \dots, n.$$

З урахуванням прийнятих позначень та припущень динаміка будь-якого вузла мережі описується в термінах “Max-plus” алгебри за допомогою рівняння:

$$x_i(k) = \tau_{ik} \otimes a_i(k) \oplus \tau_{ik} \otimes x_i(k-1). \quad (6.12)$$

Для опису динаміки телекомунікаційної мережі з  $n$  вузлів у векторній формі введемо два вектори:

$$\mathbf{a}(k) = \begin{bmatrix} a_1(k) \\ \vdots \\ a_n(k) \end{bmatrix}, \quad \mathbf{x}(k) = \begin{bmatrix} x_1(k) \\ \vdots \\ x_n(k) \end{bmatrix};$$

та діагональну матрицю обслуговування:

$$\mathbf{T}_k = \begin{bmatrix} \tau_{1k} & & \mathcal{E} \\ & \ddots & \\ \mathcal{E} & & \tau_{nk} \end{bmatrix},$$

де  $\mathcal{E} = -\infty$  – прийняте позначення в “Max-plus” алгебрі.

Тоді рівняння динаміки мережі з  $n$  вузлами у векторно-матричній формі матиме такий вид:

$$\mathbf{x}(k) = \mathbf{T}_k \otimes \mathbf{a}(k) \oplus \mathbf{T}_k \otimes \mathbf{x}(k-1). \quad (6.13)$$

При моделюванні динаміки мережі з використанням рівняння (6.13) на процеси обслуговування заявок у вузлах можна накласти деякі гіпотетичні обмеження, пов’язані з особливостями передачі інформаційних повідомлень в телекомунікаціях. Так, наприклад, можна реалізувати механізм синхронізації обробки заявок за допомогою допоміжних операцій „об’єднання” і „роз’єднання”, які виконуються в вузлах відповідно до і після обслуговування заявок [21]. Виконання операції „об’єднання” у вузлі



$i$  полягає в тому, що заявка не приєднується до черги доти, доки у вузол  $i$  не поступить хоча б по одній заявці зі всіх вузлів  $j \in P(i)$ . Тоді множина заявок  $P(i)$  об'єднується в один блок, який потім приєднується до черги заявок  $i$ -о вузла, чекаючи наступної обробки, як одне ціле.

Операція „роз'єднання” в вузлі  $i$  виконується коли завершено обслуговування чергової заявки. При цьому ця заявка замінюється на нові заявки, кількість яких дорівнює числу вузлів множини  $S(i)$ . Далі ці нові заявки одночасно покидають вузол  $i$  прямують поодинці до кожного з  $j \in S(i)$  вузлів. Для спрощення викладок припустимо, що операції „об'єднання”, „роз'єднання” та „переміщення” заявок в мережі між вузлами відбуваються миттєво.

З урахуванням особливостей обмежень щодо синхронізації, час приходу  $k$ -ї заявки до черги  $i$ -о вузла можна представити в такий спосіб:

$$a_i(k) = \begin{cases} \oplus x_j(k - r_i), & \text{якщо } P(i) \neq 0,1; \\ j \in P(i), & \text{якщо } P(i) = 1; \\ \varepsilon, & \text{якщо } P(i) = 0. \end{cases}$$

Позначивши  $M = \max\{r_i \mid r_i < \infty, i = 1, \dots, n\}$ , можна записати:

$$a_i(k) = \bigoplus_{m=0}^M \bigoplus_{j=1}^n g_{ji}^m \otimes x_j(k - m),$$

де  $g_{ij}^m$  визначається як:

$$g_{ij}^m = \begin{cases} 0, & \text{якщо } i \in P(j), \text{ та } m = r_j; \\ \varepsilon, & \text{інакше.} \end{cases}$$

Визначимо матрицю  $\mathbf{G}_m = (g_{ij}^m)$  для кожного  $m = 1, 2, \dots, M$ , яка являє собою матрицю суміжностей графа  $\mathbf{G}_m = (\mathbf{V}, \boldsymbol{\varepsilon}_m)$ , де  $\boldsymbol{\varepsilon}_m = \{(i, j) \mid i \in P(j), r_j = m\}$ .

Використовуючи матрицю  $\mathbf{G}_m$ , маємо:

$$\mathbf{a}(k) = \bigoplus_{m=0}^M \mathbf{G}_m^T \otimes \mathbf{x}(k-m),$$

де  $\mathbf{G}_m^T$  - транспонована матриця по відношенню до  $\mathbf{G}_m$ .

Підставляючи рівняння для  $\mathbf{a}(k)$  до рівняння (6.12), отримаємо:

$$\mathbf{x}(k) = \tau_k \otimes \mathbf{G}_0^T \otimes \mathbf{x}(k) \oplus \tau_k \otimes \mathbf{x}(k-1) \oplus \tau_k \otimes \bigoplus_{m=1}^M \mathbf{G}_m^T \otimes \mathbf{x}(k-m). \quad (6.14)$$

Якщо в даній моделі мережі граф, що відповідає матриці  $\mathbf{G}_0$ , є ациклічним, то рівняння (6.14) може бути розв'язане відносно  $\mathbf{x}(k)$  в явному вигляді:

$$\mathbf{x}(k) = \bigoplus_{m=1}^M \mathbf{A}_m(k) \otimes \mathbf{x}(k-m) \quad (6.15)$$

при цьому матриці  $\mathbf{A}_m$  визначаються як:

$$\begin{aligned} \mathbf{A}_1(k) &= (\mathbf{I} \oplus \mathbf{T}_k \otimes \mathbf{G}_0^T)^{\otimes r} \otimes \mathbf{T}_k \otimes (\mathbf{I} \oplus \mathbf{G}_1^T), \\ \mathbf{A}_m(k) &= (\mathbf{I} \oplus \mathbf{T}_k \otimes \mathbf{G}_0^T)^{\otimes r} \otimes \mathbf{T}_k \otimes \mathbf{G}_m^T, \\ m &= 2, \dots, M, \end{aligned}$$

де  $r$  – довжина найдовшого шляху в графі,

$\mathbf{I}$  – матриця ідентичності.

Оскільки матриця  $\mathbf{T}_k$  – діагональна, то граф, що відповідає матриці  $\mathbf{T}_k \otimes \mathbf{G}_0^T$ , є ациклічним (елементи, що дорівнюють  $\varepsilon$  в матриці  $\mathbf{G}_0^T$  дорівнюють  $\varepsilon$  в матриці  $\mathbf{T}_k \otimes \mathbf{G}_0^T$ ). Тоді можемо прийняти, що:

$$\mathbf{A} = \mathbf{T}_k \otimes \mathbf{G}_0^T;$$

і вектор:

$$\mathbf{b} = \mathbf{T}_k \otimes \mathbf{x}(k-1) \oplus \mathbf{T}_k \otimes \bigoplus_{m=1}^M \mathbf{G}_m^T \otimes \mathbf{x}(k-m).$$

З урахуванням позначень отримуємо:

$$\begin{aligned}
 \mathbf{x}(k) &= (\mathbf{I} \otimes \mathbf{A})^{\otimes r} \otimes \mathbf{b} = \\
 &= (\mathbf{I} \oplus \mathbf{T}_k \otimes \mathbf{G}_0^T)^{\otimes r} \otimes (\mathbf{T}_k \otimes \mathbf{x}(k-1) \oplus \mathbf{T}_k \otimes \bigoplus_{m=1}^M \mathbf{G}_m^T \otimes \mathbf{x}(k-m)) = \\
 &= (\mathbf{I} \oplus \mathbf{T}_k \otimes \mathbf{G}_0^T)^{\otimes r} \otimes ((\mathbf{T}_k \oplus \mathbf{G}_1^T) \otimes \mathbf{x}(k-1) \oplus \mathbf{T}_k \otimes \bigoplus_{m=2}^M \mathbf{G}_m^T \otimes \mathbf{x}(k-m)) = \\
 &= \bigoplus_{m=1}^M \mathbf{A}_m(k) \otimes \mathbf{x}(k-m).
 \end{aligned}$$

Для прикладу розглянемо модель мережі, зображену на рис. 6.8 (в фігурних дужках вказана кількість заявок в вузлах на початковий момент часу).

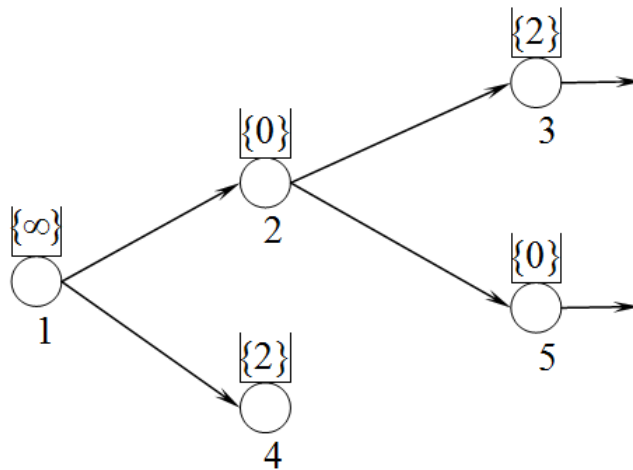


Рисунок 6.8 – Модель ділянки телекомунікаційної мережі

Неявне рівняння (6.15) для цієї моделі матиме вигляд:

$$\mathbf{x}(k) = \mathbf{T}_k \otimes (\mathbf{G}_0^T \otimes \mathbf{x}(k) \oplus \otimes \mathbf{x}(k-1) \oplus \otimes \mathbf{G}_2^T \otimes \mathbf{x}(k-2)),$$

де

$$\mathbf{G}_0 = \begin{bmatrix} \varepsilon & 0 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & 0 \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{bmatrix}, \quad \mathbf{G}_2 = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & 0 & \varepsilon \\ \varepsilon & \varepsilon & 0 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{bmatrix}.$$

Звідси це рівняння може бути розв'язане відносно  $\mathbf{x}(k)$  в явному вигляді:

$$\mathbf{x}(k) = \mathbf{A}_1(k) \otimes \mathbf{x}(k-1) \oplus \mathbf{A}_2(k) \otimes \mathbf{x}(k-2),$$

де:

$$\mathbf{A}_1 = \begin{bmatrix} \tau_{1k} & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \tau_{1k} \otimes \tau_{2k} & \tau_{2k} & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \tau_{3k} & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \tau_{4k} & \varepsilon \\ \tau_{1k} \otimes \tau_{2k} \otimes \tau_{5k} & \tau_{2k} \otimes \tau_{5k} & \varepsilon & \varepsilon & \tau_{5k} \end{bmatrix},$$

$$\mathbf{A}_2 = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \tau_{1k} \otimes \tau_{2k} & \tau_{2k} & \varepsilon & \varepsilon & \varepsilon \\ \tau_{1k} & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{bmatrix}.$$

Розглянемо мережу, аналогічну представлений на рис. 6.8, з тією різницею, що в початковий момент часу всі обслуговуючі пристрої мережі вільні. Черга заявок в кожному вузлі-джерелі має безконечну довжину, а черги всіх інших вузлів не містять заявок.

Тоді, з умови ациклічності графа, динаміка мережі може бути представлена за допомогою рівняння (6.15) та матиме такий вигляд:

$$\mathbf{x}(k) = \mathbf{A}(k) \otimes \mathbf{x}(k-1) \tag{6.16}$$

де:

$$\mathbf{A}(k) = (\mathbf{I} \oplus \mathbf{T}_k \otimes \mathbf{G}^T)^{\otimes r} \otimes \mathbf{T}_k, \tag{6.17}$$

$\mathbf{G} = (g_{ij})$  – матриця, елементи якої визначаються в такий спосіб:

$$g_{ij} = \begin{cases} 0, & \text{якщо } i \in P(j); \\ \varepsilon, & \text{інакше,} \end{cases}$$

а  $r$  – довжина найдовшого шляху в графі мережі  $\mathbf{G}$ .

Для прикладу розглянемо ділянку мережі з  $n = 5$  вузлами (рис. 6.9). На рисунку не вказується кількість заявок в вузлах в початковий момент часу, оскільки вважається, що спочатку у вузлі-джерелі міститься нескінченний потік заявок, а в проміжних вузлах заявки відсутні. Матриця  $\mathbf{G}$ , що відповідає цій мережі, матиме вигляд:

$$\mathbf{G} = \begin{bmatrix} \varepsilon & 0 & 0 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & 0 & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & 0 & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & 0 \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{bmatrix}.$$

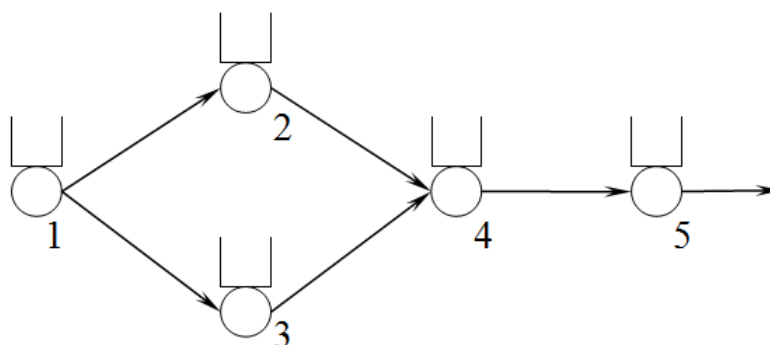


Рисунок 6.9 – Структура ділянки телекомунікаційної мережі

З огляду на те, що для графа мережі довжина найбільшого шляху  $r = 3$ , рівняння (6.16) матиме вигляд:

$$\mathbf{x}(k) = (\mathbf{I} \oplus \mathbf{T}_k \otimes \mathbf{G}^T)^{\otimes 3} \otimes \mathbf{T}_k \otimes \mathbf{x}(k-1),$$

а матриця (6.17) прийме вигляд:

$$A(k) = \begin{bmatrix} \tau_{1k} & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \tau_{1k} \otimes \tau_{2k} & \tau_{2k} & \varepsilon & \varepsilon & \varepsilon \\ \tau_{1k} \otimes \tau_{3k} & \varepsilon & \tau_{3k} & \varepsilon & \varepsilon \\ \otimes_{i=1}^4 \tau_{ik} & \tau_{2k} \otimes \tau_{4k} & \tau_{3k} \otimes \tau_{4k} & \tau_{4k} & \varepsilon \\ \otimes_{i=1}^5 \tau_{ik} & \tau_{2k} \otimes \tau_{3k} \otimes \tau_{4k} & \tau_{3k} \otimes \tau_{4k} \otimes \tau_{5k} & \tau_{4k} \otimes \tau_{5k} & \tau_{5k} \end{bmatrix}.$$

При застосуванні вищезазначеного рівняння можна визначити середній час робочого циклу ділянки мережі.

Скористаємося оператором, який є спеціальним для “Max-plus” і означає максимальний елемент матриці  $\mathbf{A}$ :

$$\text{для } \forall \mathbf{A} = \{a_{ij}\}_{i,j=1}^m \quad \|\mathbf{A}\| = \bigoplus_{ij} a_{ij}.$$

Припустимо час закінчення  $k$ -о циклу буде  $\|\mathbf{x}(k)\|$ . Інакше кажучи,  $k$ -й робочий цикл мережі завершується після закінчення обслуговування  $k$ -х вимог в усіх вузлах даної мережі.

Можна сформулювати задачу про знаходження величини  $\lim_{k \rightarrow \infty} \otimes^k \sqrt{\|\mathbf{x}(k)\|} = \lim_{k \rightarrow \infty} \otimes^k \sqrt{\|\mathbf{A}(k)\|}$ , що являє собою середній час робочого циклу.

Для (6.16) маємо:

$$\otimes^k \sqrt{\|\mathbf{x}(k)\|} = \otimes^k \sqrt{\|\mathbf{A}(k) \otimes \mathbf{x}(k-1)\|} = \otimes^k \sqrt{\|\mathbf{A}(k) \otimes \dots \otimes \mathbf{A}(1) \otimes \mathbf{x}(0)\|}.$$

Введемо позначення:

$$\mathbf{A}_k = \mathbf{A}(k) \otimes \dots \otimes \mathbf{A}(1), \quad (6.18)$$

та отримаємо:

$$\otimes^k \sqrt{\|\mathbf{x}(k)\|} = \otimes^k \sqrt{\|\mathbf{A}_k\|}. \quad (6.19)$$

Дійсно, (6.19) витікає з того, що

$$\begin{aligned} \|\mathbf{x}(k)\| &= \bigoplus_i x_i(k) = \bigoplus_i \bigoplus_j \{\mathbf{A}_k\}_{ij} \otimes x_j(0) = \\ &= \bigoplus_i \bigoplus_j \{\mathbf{A}_k\}_{ij} \otimes 0 = \bigoplus_{ij} \{\mathbf{A}_k\}_{ij} = \|\mathbf{A}_k\| \end{aligned}$$

Тобто, постає задача про знаходження оцінки величини:

$$\lim_{k \rightarrow \infty} \otimes^k \sqrt{\|\mathbf{x}(k)\|} = \lim_{k \rightarrow \infty} \otimes^k \sqrt{\|\mathbf{A}(k)\|}. \quad (6.20)$$

Звісно [21], що середній час робочого циклу можна оцінити в такий спосіб:

$$\|E[\mathbf{T}]\| \leq \lim_{k \rightarrow \infty} \otimes^k \sqrt{\|\mathbf{x}(k)\|} \leq E\|\mathbf{T}\|. \quad (6.21)$$

Також, в роботі [15] доведено, що точне значення середнього часу робочого циклу дорівнює нижній границі (6.21) для мереж ланцюгової структури (рис. 6.10):

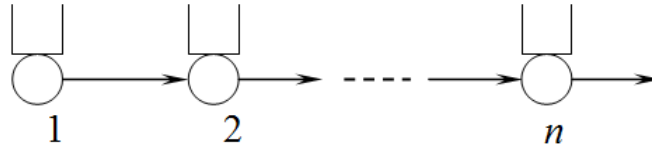


Рисунок 6.10 – Ланцюгова структура мережі

Отже, для середнього часу робочого циклу мережі ланцюгової структури справедливо рівняння:

$$\lim_{k \rightarrow \infty} \otimes^k \sqrt{\|\mathbf{x}(k)\|} = E\|\mathbf{T}\|. \quad (6.22)$$

## 7 Аналіз асимптотичної поведінки матриці $\mathbf{A}^k$ (частотні властивості ДБС)

В цьому розділі розглянуто аналог відомої теорії Перрона-Фробеніуса [71] в частинні застосування особливостей представлення моделей ДБС за допомогою Max-Plus алгебри. Предметом розгляду є поведінка системи, якій відповідає матриця  $\mathbf{A}$ , на значному часовому інтервалі спостереження. Цікавими є питання зміни властивостей матриці  $\mathbf{A}^k$  при великому значенні  $k$ .

Приймаючи до уваги, що  $\mathbf{A}$  є  $n \times n$  матриця, яка визначена в  $R_{\max}$ . Для зручності введено обмеження на структуру графа  $G(\mathbf{A})$ . Припустимо, що в графі є щонайменш один замкнутий контур, в іншому випадку матриця  $\mathbf{A}$  є нільпотентною і  $\mathbf{A}^k = \mathbf{e}$  для досить великих  $k$ . Також будемо вважати, що вага замкнутого контуру дорівнює  $e$ . Якщо це не так, то матрицю  $\mathbf{A}$  можна нормалізувати шляхом розділення всіх її елементів на величину власного числа матриці  $\lambda$ . В цьому випадку поведінка системи може бути описана формулою перетворень:

$$\mathbf{A}^k = \lambda^k (\lambda^{-1} \mathbf{A})^k. \quad (7.1)$$

В такому представленні ДБС завідомо має контури з від'ємною вагою, причому деякі з них матимуть вагу, що дорівнює  $e$ . Це означає, що для нормалізованої матриці  $\mathbf{A}$  власне число завжди дорівнює  $e$ .

Для аналізу частотних властивостей ДБС досить широко застосовуються особливі характеристики і поняття, які в загальному випадку мають назву критичних графів нормалізованої матриці  $\mathbf{A}$  [63]. Зокрема, замкнутий контур  $\xi$  графа  $G(\mathbf{A})$  називається критичним, якщо



він має максимальну вагу  $|\xi|_w = e$ . Граф  $G^C(\mathbf{A})$  називається критичним, якщо він має тільки ті вершини і дуги  $G(\mathbf{A})$ , які належать критичному контуру цього графа. Множина вершин критичного контуру позначається  $v^C$ . Насиченим називають граф  $S(\mathbf{A}, \mathbf{y})$ , який містить вершини і дуги графа  $G(\mathbf{A})$ , для яких  $A_{ij}y_j = y_i$ , причому  $y_i, y_j \neq \varepsilon$ . Циклічність максимально повнозв'язного підграфа – це найбільший спільний дільник (НСД) довжин всіх його контурів, а циклічність  $C(G)$  графа  $G(\mathbf{A})$  – найменше спільне кратне (НСК) циклічностей всіх його підграфів.

Вищенаведені поняття і визначення можна проілюструвати на прикладах. Програми обчислень наведені у додатку, лістинг 4.

Приклад 7.1 Обчислення циклічності графа.

Припустимо існує нормалізована матриця  $\mathbf{A}$ :

$$\mathbf{A} = \begin{bmatrix} e & e & \varepsilon & \varepsilon \\ -1 & -2 & \varepsilon & \varepsilon \\ \varepsilon & -1 & -1 & e \\ \varepsilon & \varepsilon & e & e \end{bmatrix}.$$

Граф  $G(\mathbf{A})$  для цієї матриці має такий вигляд.:

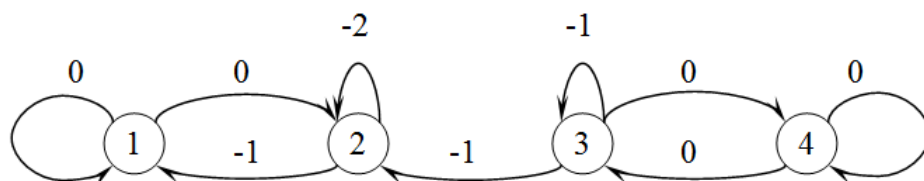


Рисунок 7.1 – Структура графа  $G(\mathbf{A})$

Граф  $G(\mathbf{A})$  має три критичні конури –  $\{1\}, \{3,4\}, \{4\}$ .

Критичному графу  $G^C(\mathbf{A})$  відповідає така матриця  $\mathbf{C}$ :

$$C = \begin{bmatrix} e & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & e \\ \varepsilon & \varepsilon & e & e \end{bmatrix}.$$

Матриця  $A$  має власний вектор  $y = [e \quad -1 \quad -2 \quad -2]$ , що відповідає власному числу  $\lambda = e$ . Тоді граф насичення  $S$  для матриці  $A$  буде мати вигляд:

$$S = S(A, y) = \begin{bmatrix} e & \varepsilon & \varepsilon & \varepsilon \\ -1 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & -1 & \varepsilon & e \\ \varepsilon & \varepsilon & e & e \end{bmatrix}.$$

Циклічність графа  $G(A)$  може бути оцінена у такий спосіб.

Критичний граф  $G^C(A)$  має два жорстко повнозв'язні підграфи з вершинами  $\{1\}$  і  $\{3,4\}$  відповідно. Другий підграф має два критичні контури  $\{4\}$  і  $\{3,4\}$  довжиною 1 і 2 відповідно. Циклічність першого підграфа дорівнює 1, циклічність другого також 1, тому що НСД(1, 2) = 1. Тоді циклічність  $G^C(A) \in$  НСК(1, 1) = 1.

Узагальнюючи ці прості результати у наведених прикладах, можна сформулювати наступні твердження, що мають важливе значення при розгляді процесів у ДБС. Так, будь-який контур критичного графа  $G^C(A)$  є критичним. Також, якщо взяти будь-яку пару вершин матриці  $G^C(A)$ , то шляхи, які з'єднують вершини  $i$  та  $j$  у критичному графі мають однакову вагу.

Для аналізу частотних властивостей ДБС представляє інтерес множина власних векторів матриці  $A$ , пов'язаних з її власним числом, яке для нормалізованого представлення завжди дорівнює  $e$ . Ця множина має

назву модулятора частотних властивостей ДБС. В цьому випадку множина стовпців власних векторів матриці  $\mathbf{A}$  розглядається як генератор цього модулятора і позначається як матриця  $\mathbf{A}^+$ , для якої є справедливими такі твердження:

– якщо  $\mathbf{y}$  – власний вектор матриці  $\mathbf{A}$ , то він також є власним вектором матриці  $\mathbf{A}^+$ , причому це є лінійна комбінація стовпців  $(\mathbf{A}^+)_{\bullet i}$ , де  $i \in \nu^C$ , а саме:

$$\mathbf{y} = \bigoplus_{i \in \nu^C} y_i (\mathbf{A}^+)_{\bullet i} \quad (7.2)$$

– якщо найдовший контур матриці  $\mathbf{A}$  має вагу (власне число)  $e$ , то будь-який власний вектор, пов'язаний з цим числом, є лінійною комбінацією з  $N_{\mathbf{A}}^C$  стовпців матриці  $\mathbf{A}^+$ . В цьому трактуванні  $N_{\mathbf{A}}^C$  – це число підграфів в критичному графі  $G^C(\mathbf{A})$ .

З вище наведених тверджень витікають наслідки, які мають практичне застосування:

– всі стовпці матриці  $\mathbf{A}^+$  є власними векторами матриці  $\mathbf{A}$ , тобто  $(\mathbf{A}^+)_{\bullet i}$ ,  $i \in \nu^C$  – власні вектори;

– для вузлів  $i$  та  $j$ , що належать одному й тому ж підграфу критичного графу  $G^C(\mathbf{A})$ , справедливим є твердження про пропорційність  $(\mathbf{A}^+)_{\bullet i}$  та  $(\mathbf{A}^+)_{\bullet j}$ ;

– жоден з власних векторів (стовпців матриці  $(\mathbf{A}^+)_{\bullet i}$ ) не може бути представлений як лінійна комбінація стовпців  $(\mathbf{A}^+)_{\bullet j}$ , які одержані тільки з тих вузлів  $j$ , що належать до множини  $\nu^C$  підграфа  $G^C(\mathbf{A})$ , відмінних від  $i$ .

Ці практичні висновки можна проілюструвати прикладом.

Приклад 7.2

Для нормалізованої матриці  $\mathbf{A}$  графа  $G(\mathbf{A})$  (рис. 7.1) відповідна матриця  $\mathbf{A}^+$  має вид:

$$\mathbf{A}^+ = \begin{bmatrix} e & e & \varepsilon & \varepsilon \\ -1 & -1 & \varepsilon & \varepsilon \\ -2 & -1 & e & e \\ -2 & -1 & e & e \end{bmatrix}.$$

Вузли 1, 3, 4 належать до критичного графу  $G^C(\mathbf{A})$ , в якому є два підграфи, зокрема  $\{1\}$  і  $\{3,4\}$ . Стівці 1 і 3 є незалежними власними векторами матриці  $\mathbf{A}$ . Стівець 4 є еквівалентним стівцю 3.

Для нормалізованої матриці  $\mathbf{A}$  графа  $G(\mathbf{A})$  (рис. 7.1) також може бути визначена частотна проекція. Частотна проекція  $\mathbf{A}$  визначається як деяка матриця  $\mathbf{Q}$ , яка задовольняє наступним умовам:

$$\mathbf{A}\mathbf{Q} = \mathbf{Q}\mathbf{A} = \mathbf{Q}^2. \quad (7.3)$$

З урахуванням властивостей матриці  $\mathbf{A}^+$  можна стверджувати, що матриця:

$$(\mathbf{Q})_i = (\mathbf{A}^+)_{\bullet i} (\mathbf{A}^+)_{i \bullet}, \quad i \in V^C \quad (7.4)$$

є складовою частиною частотної проекції матриці  $\mathbf{A}$ . Тоді повна проекція:

$$\mathbf{Q} = \bigoplus_{i \in V^C} (\mathbf{Q})_i \quad (7.5)$$

Приклад 7.3

Для раніше розглянутого прикладу (рис. 7.1) елементарні складові та повна частотна проекція матриці  $\mathbf{A}$  визначається в такий спосіб:

$$(\mathbf{Q})_1 = \begin{bmatrix} e \\ -1 \\ -2 \\ -2 \end{bmatrix} \cdot [e \quad e \quad \varepsilon \quad \varepsilon] = \begin{bmatrix} e & e & \varepsilon & \varepsilon \\ -1 & -1 & \varepsilon & \varepsilon \\ -2 & -2 & \varepsilon & \varepsilon \\ -2 & -2 & \varepsilon & \varepsilon \end{bmatrix},$$

$$(\mathbf{Q})_{3,4} = \begin{bmatrix} \varepsilon \\ \varepsilon \\ e \\ e \end{bmatrix} \cdot [-2 \quad -1 \quad e \quad e] = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ -2 & -1 & e & e \\ -2 & -1 & e & e \end{bmatrix},$$

$$\mathbf{Q} = (\mathbf{Q})_1 \oplus (\mathbf{Q})_{3,4} = \begin{bmatrix} e & e & \varepsilon & \varepsilon \\ -1 & -1 & \varepsilon & \varepsilon \\ -2 & -1 & e & e \\ -2 & -1 & e & e \end{bmatrix}.$$

З урахуванням вище наведених визначень і зауважень можна сформулювати необхідні і достатні умови збіжності степенів матриці  $\mathbf{A}$ . Існують різні підходи при розгляді цього питання. Найбільш повно ці умови сформульовані в [71] і полягають в наступному:

- Для матриці  $\mathbf{A}$  розмірності  $n \times n$  при  $n \rightarrow \infty$

$$\lim_{k \rightarrow \infty} \mathbf{A}^k = \mathbf{Q}, \tag{7.6}$$

за умови, що циклічність будь-якого з підграфів критичного графу  $G^C(\mathbf{A})$  дорівнює 1.

- Якщо  $n$  має строго визначене скінчене значення ( $n < \infty$ ), то існує таке число  $K$ , що  $\forall k \geq K: \mathbf{A}^k = \mathbf{Q}$  за умови, що циклічність кожного з підграфів графу  $G^C(\mathbf{A})$  дорівнює 1.

Приклад 7.4

Нехай 
$$\mathbf{A} = \begin{bmatrix} e & e & \varepsilon & \varepsilon \\ -1 & -2 & \varepsilon & \varepsilon \\ \varepsilon & -1 & -1 & e \\ \varepsilon & \varepsilon & e & e \end{bmatrix},$$
 тоді 
$$\mathbf{A}^2 = \begin{bmatrix} e & e & \varepsilon & \varepsilon \\ -1 & -1 & \varepsilon & \varepsilon \\ -2 & -2 & e & e \\ \varepsilon & -1 & e & e \end{bmatrix},$$

$$\mathbf{A}^3 = \mathbf{A}^4 = \begin{bmatrix} e & e & \varepsilon & \varepsilon \\ -1 & -1 & \varepsilon & \varepsilon \\ -2 & -1 & e & e \\ -2 & -1 & e & e \end{bmatrix}.$$

При  $k \geq 3$ ,  $\mathbf{A}^k = \mathbf{Q}$ . Матриця  $\mathbf{Q}$  була розрахована в попередньому прикладі 7.3.

В наведеному прикладі періодичний режим в ДБС досягається за скінчене число кроків. Це завжди справедливо для підграфу з вузлами, що належать до критичного графу  $G^C(\mathbf{A})$ , але невірно для будь-якої матриці  $\mathbf{A}$ . Припустимо нормалізована матриця  $\mathbf{A}$  має вигляд:

$$\mathbf{A} = \begin{bmatrix} -1 & \varepsilon \\ \varepsilon & e \end{bmatrix}.$$

В цій матриці  $\varepsilon$  умовний вхід, який наближається до  $\varepsilon$ . Коли всі входи збігаються до кінцевого значення, періодичний режим може бути досягнутий, але час, необхідний для цього, може бути скільки завгодно великим.

Якщо взяти матрицю  $\mathbf{A}$  наступного виду:

$$\mathbf{A} = \begin{bmatrix} -r & -1 \\ e & e \end{bmatrix},$$

за умови, що  $r$  відносно невелике число, то матриця  $\mathbf{A}^k$  збігається до такої матриці:

$$\mathbf{A}^k = \begin{bmatrix} -1 & -1 \\ e & e \end{bmatrix},$$

але це досягається лише за  $k = 1/r$  кроків.

У загальному випадку частотна теорія ДБС може бути представлена з використанням поняття циклічності матриці. Відомо [58], що матриця  $\mathbf{A}$  є абсолютно циклічною, якщо існують такі  $d$  і  $M$ , що  $\forall m \geq M : \mathbf{A}^{m+d} = \mathbf{A}^m$ , при цьому  $d$  називають показником циклічності матриці  $\mathbf{A}$ , а саму матрицю називають  $d$ -циклічною.

Також відомо [58], що матриця  $\mathbf{A}$  є асимптотично циклічною, якщо існує таке  $d$ , та для всіх  $\theta \geq 0$  існує таке  $M$ , що для всіх  $m \geq M$ ,  $\sup_{ij} |(\mathbf{A}^{m+d})_{ij} - (\mathbf{A}^m)_{ij}| \leq \theta$ .

У цьому випадку  $d$  називають показником асимптотичної циклічності матриці  $\mathbf{A}$ , а саму матрицю називають  $d$ -асимптотично циклічною.

Звідси справедливим є твердження, що будь-яка матриця є асимптотично циклічною, якщо в графі є хоча б один замкнутий контур. Показник асимптотичної циклічності  $d$  матриці  $\mathbf{A}$  дорівнює показнику циклічності  $\rho$  критичного графу  $G^C(\mathbf{A})$  матриці  $\mathbf{A}$ . Крім того, якщо  $G(\mathbf{A})$  і  $G(\mathbf{A}^\rho)$  зв'язні, то матриця  $\mathbf{A}$  має показник циклічності  $\rho$  [71].

## 8 Представлення графа синхронізації ДБС в термінах мережі Петрі за допомогою “Max-Plus” алгебри

Поняття графа синхронізації дискретно-безперервних систем по відношенню до управління процесами вперше було введено в теорії моделювання динаміки поведінки ДБС за допомогою апарату мереж Петрі [61]. Але аналітичний опис дискретно-безперервних процесів за правилами мереж Петрі є досить громіздким і не є прийнятним в рамках підходів, які використовуються в сучасній теорії управління. При застосуванні апарату Max-Plus-алгебри можна використовувати векторно-матричні рівняння, які дозволяють формалізувати представлення ДБС в формі, що подібна моделям динаміки в просторі змінних стану.

### 8.1 Загальний підхід до вирішення задачі

В загальному випадку типова структура переходу графа синхронізації ДБС може бути представлена у такому вигляді:

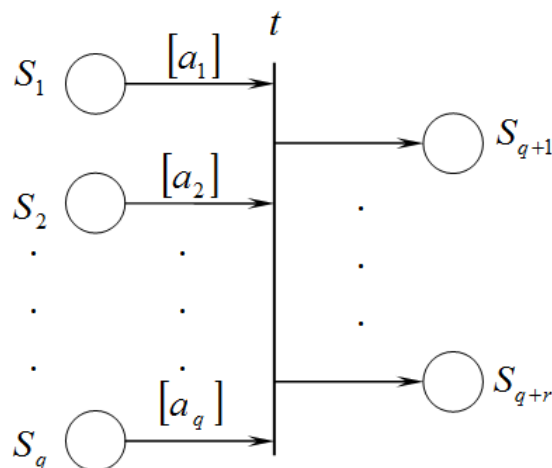


Рисунок 8.1 – Структура переходу графа синхронізації ДБС



В даному випадку розглядається перехід з одного стану в інший, який має назву некерованого. Правило переходу в інший стан або спрацювання некерованого переключення відбувається лише тоді, коли всі положення від  $S_i$  до  $S_q$  в передобласті переходу будуть зайняті так довго, як це задається часовими оцінками сповільнення процесу  $a_i$ . Дискретний момент переключення переходу  $x_j$  можна розглядати як аналог часу маркування переходу  $S_j$  в моделях мережі Петрі. Кожне  $x_j$  ( $j = q + 1, \dots, q + r$ ) в післяобласті переходу визначається як максимальне значення маркованих точок від  $x_1$  до  $x_q$  в передобласті переходу з урахуванням відповідного часу сповільнення процесу, а саме:

$$x_j = \bigoplus_{i=1}^q a_i x_i, \quad j = q + 1, \dots, q + r \quad (8.1)$$

Якщо в графі розглядається керований із зовні перехід, то прийнято вважати, що він відбувається при додатковій логічній умові переходу, яка задається апріорі. Часова відмітка, в якій логічна умова є дійсною, позначається  $u$  і тоді для маркованих часових точок стану ДБО в післяобласті керованого переходу маємо:

$$x_j = \left[ \bigoplus_{i=1}^q a_i x_i \right] \oplus u, \quad j = q + 1, \dots, q + r \quad (8.2)$$

Таким чином, для кожного переміщення в графі синхронізації ДБО можна встановити часові моменти переходу і марковані часові відмітки заняття стану в післяобласті.

Розглянутий підхід для окремого переходу можна розповсюдити на всю послідовність переходів графа синхронізації ДБС.

Припустимо, що є деякий граф синхронізації системи з  $S$  положеннями і  $P$  логічними зовнішніми умовами переключення

переходів. Використовуючи підхід для окремого переходу (8.1, 8.2) для графа синхронізації можна отримати систему з  $n$  рівнянь:

$$\begin{aligned}x_1 &= a_{11}x_1 \oplus \dots \oplus a_{1n}x_n \oplus b_{11}u_1 \oplus \dots \oplus b_{1p}u_p, \\ &\vdots \\ x_n &= a_{n1}x_1 \oplus \dots \oplus a_{nn}x_n \oplus b_{n1}u_1 \oplus \dots \oplus b_{np}u_p.\end{aligned}\tag{8.3}$$

Або в векторно-матричній формі вона може бути представлена у вигляді:

$$\mathbf{x} = \mathbf{A} \mathbf{x} \oplus \mathbf{B} \mathbf{u},\tag{8.4}$$

де:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

– вектор стану, кожний елемент  $x_i$  якого фіксує момент часу маркування (включення) переходу  $S_i$ .

Вплив зовнішніх логічних умов маркування (переключення) задається вектором керування:

$$\mathbf{u} = \begin{bmatrix} u_1 \\ \vdots \\ u_p \end{bmatrix}.$$

Кожна логічна умова маркування (переключення) переходу  $S_i$  стає дійсною в деякій часовій точці розвитку процесу, яка також враховується в рівняннях (8.3). В цьому сенсі кожний окремий елемент  $u_i$ ,  $i = 1, \dots, p$  вектора керування управління розглядається як управляючий вплив на  $i$ -й перехід.

Значення кожного елемента  $a_{ij}$  матриці **A** рівняння (8.4) відповідає часовій оцінці передобласті переходу  $T_i(S_i, t_k)$  з положення  $S_i$  в положення  $S_j$ , тобто характеризує часову інерційність переходу. Формально це можна представити наступним виразом:

$$a_{ij} = \begin{cases} T_i(S_i, t_k), & \text{для випадку } \exists t_k : S_i \in t_k \bullet \wedge S_j \in \bullet t_k, \\ 0, & \text{інакше.} \end{cases} \quad (8.5)$$

Якщо на переміщення  $t_k$  додатково впливає логічна умова переключення, наприклад  $u_r$ , то елемент матриці **B** дорівнює 0, в іншому випадку  $b_{ir} = \varepsilon$ .

Формально це можна представити наступним чином:

$$b_{ir} = \begin{cases} 0, & \text{для випадку } \exists t_k : S_i \in t_k \bullet \text{ і } u_r, \text{ що відповідає } t_k, \\ \varepsilon, & \text{інакше.} \end{cases}$$

Елементи матриці **B** таким чином вказують на ті переходи ДБС, які є керованими зовні.

В графах синхронізації переходи з одного положення в інше створюють замкнені цикли. Тобто, через певне число переходів умовний маркер проходить всі позиції графа і виходить на повторне маркування тих самих позицій. Для того, щоб відрізнити окремі цикли поведінки ДБС, всі змінні вектора стану  $x$  і вектора управління  $u$  мають індекс  $k$ , який показує з якою частотою буде маркуватися відповідне положення в графі синхронізації. Тобто,  $x_i(k)$  – це часова точка в якій положення  $S_i$  займається (маркується)  $k$ -й раз.

Для встановлення початку нового переміщення по циклу має бути визначено положення  $S_i$ , зайняття (маркування) якого означає, що закінчено попередній цикл  $k$  і починається новий  $(k+1)$  цикл переміщення по графу синхронізації. Зазвичай, для вибору часової точки

початку нового циклу на графі задають початкову (стартову) позицію, відносно якої і ведуть відлік початку нового циклу. В цьому випадку в термінах змінних стану говорять про розрахунок стану  $x_i(k+1)$  через  $x_i(k)$ . Якщо ж положення  $S_i$  не є стартовою позицією циклу, то  $x_i(k+1)$  знаходиться через  $x_j(k+1)$ , тобто відносно часової відмітки в поточному циклі. Наведені ствердження для окремої стартової позиції графа синхронізації на рис. 8.1 матимуть наступне представлення з урахуванням впливу зовнішнього логічного управління  $u$ :

$$x_j(k+1) = a_1 x_1(k) \oplus \left( \bigoplus_{i=2}^{q-1} a_j x_j(k+1) \right) \oplus a_q x_q(k) \oplus u(k+1) \quad (8.7)$$

Індекс зовнішнього управління прийнято таким самим, як індекс змінних стану відповідного циклу.

Як приклад розглянемо граф синхронізації, що має структуру представлену на рисунку 8.2.

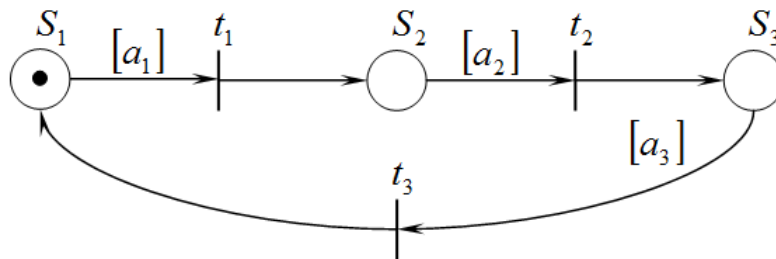


Рисунок 8.2 – Простий граф синхронізації

В якості стартової позиції початку циклу умовно взята позиція  $S_2$ .

Тоді система рівнянь зі змінними стану матиме вигляд:

$$\begin{aligned} x_1(k+1) &= a_3 x_3(k+1), \\ x_2(k+1) &= a_1 x_1(k), \\ x_3(k+1) &= a_2 x_2(k+1). \end{aligned}$$

В матричному вигляді ця система може бути представлена наступним чином:

$$\mathbf{x}(k+1) = \mathbf{A}_0 \mathbf{x}(k+1) \oplus \mathbf{A}_1 \mathbf{x}(k) \oplus \mathbf{B} \mathbf{u}(k+1) \quad (8.8)$$

В залежності від початкового маркування позиції  $S_i$  матриця  $\mathbf{A}$  таким чином розбивається на дві  $\mathbf{A}_0$  і  $\mathbf{A}_1$ , причому

$$\mathbf{A} = \mathbf{A}_0 \oplus \mathbf{A}_1 \quad (8.9)$$

Матриці  $\mathbf{A}_0$  і  $\mathbf{A}_1$  визначаються із матриці  $\mathbf{A}$  за допомогою матриць трансформації  $\mathbf{T}_{\mathbf{A}_0}$  і  $\mathbf{T}_{\mathbf{A}_1}$ .

$$\mathbf{A}_0 = \mathbf{A} \cdot \mathbf{T}_{\mathbf{A}_0} \quad (8.10)$$

$$\mathbf{A}_1 = \mathbf{A} \cdot \mathbf{T}_{\mathbf{A}_1} \quad (8.11)$$

Матриці трансформації враховують різні початкові маркування позиції  $S_i$  наступним чином:

$$\left(\mathbf{T}_{\mathbf{A}_0}\right)_{ij} = \begin{cases} e, & \text{якщо } i = j \text{ і положення } S_i \text{ не марковане,} \\ \varepsilon, & \text{в усіх інших випадках.} \end{cases} \quad (8.12)$$

$$\left(\mathbf{T}_{\mathbf{A}_1}\right)_{ij} = \begin{cases} e, & \text{якщо } i = j \text{ і положення } S_i \text{ марковане,} \\ \varepsilon, & \text{в усіх інших випадках.} \end{cases} \quad (8.13)$$

З визначення матриць (8.12) і (8.13) витікає, що тільки елементи головної діагоналі матриць трансформації можуть бути більшими за  $\varepsilon$ . Сума обох матриць завжди дає одиничну матрицю.

Система рівнянь (8.8) може бути розв’язана відомими методами [37]:

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}_0 \mathbf{x}(k+1) \oplus \mathbf{A}_1 \mathbf{x}(k) \oplus \mathbf{B} \mathbf{u}(k+1) = \\ &= \mathbf{A}_0^2 \mathbf{x}(k+1) \oplus \mathbf{A}_0 \mathbf{A}_1 \mathbf{x}(k) \oplus \mathbf{A}_0 \mathbf{B} \mathbf{u}(k+1) \oplus \mathbf{A}_1 \mathbf{x}(k) \oplus \mathbf{B} \mathbf{u}(k+1) = \\ &\quad \vdots \\ &= \mathbf{A}_0^n \mathbf{x}(k+1) \oplus [\mathbf{A}_0^{n-1} \oplus \dots \oplus \mathbf{A}_0 \oplus \mathbf{I}] \mathbf{A}_1 \mathbf{x}(k) \oplus [\mathbf{A}_0^{n-1} \oplus \dots \oplus \mathbf{A}_0 \oplus \mathbf{I}] \mathbf{B} \mathbf{u}(k+1). \end{aligned} \quad (8.14)$$

Очевидно, що підграф  $G(\mathbf{A}_0)$  загального графа синхронізації не має внутрішніх циклів, максимальна довжина шляху в цьому графі дорівнює  $n-1$ , і тому  $\mathbf{A}_0^n = \mathbf{N}$ , тобто завжди має скінчене значення.

За допомогою матриці:

$$\mathbf{A}_0^* = \mathbf{I} \oplus \mathbf{A}_0 \oplus \mathbf{A}_0^2 \oplus \dots \oplus \mathbf{A}_0^{n-1} \quad (8.15)$$

і позначення:

$$\mathbf{M} = \mathbf{A}_0^* \mathbf{A}_1 \quad (8.16)$$

рівняння (8.14) спрощується:

$$\mathbf{x}(k+1) = \mathbf{M}\mathbf{x}(k) \oplus \mathbf{A}_0^* \mathbf{B}\mathbf{u}(k+1). \quad (8.17)$$

В цьому випадку матриця  $\mathbf{M}$  може розглядатися як матриця динаміки системи без зовнішнього керування – динамічна характеристика вільної поведінки системи:

$$\mathbf{x}(k+1) = \mathbf{M}\mathbf{x}(k). \quad (8.18)$$

Тобто, динамічні характеристики ДБС залежать від структури і властивостей матриці  $\mathbf{M}$  відповідно до рівняння (8.18).

## 8.2 Структура і властивості матриці динаміки замкнутої ДБС

Відносно структури матриці  $\mathbf{M}$  можна стверджувати, що початково марковані позиції графа синхронізації завжди розглядаються як вершини графа  $G(\mathbf{M})$ , які мають тільки вихідні ребра. Тому очевидно, що для кожного початково маркованого положення  $S_i$  графа синхронізації в матриці  $\mathbf{A}_1$  є хоча б один елемент в стовпці  $(\mathbf{A}_1)_{\cdot j}$ , відмінний від  $\varepsilon$ , що говорить про те, що існує наступне положення  $S_k$ , в яке обов'язково переходить система з положення  $S_i$ . Якщо ж положення  $S_i$  не є початково

маркованим, то  $j$ -й стовпець матриці  $(\mathbf{A}_1)_{\bullet j}$  має тільки елементи  $\varepsilon$ . Таким чином, має силу наступне співвідношення:

$$\begin{aligned} (\mathbf{A}_0^* \mathbf{A}_1)_{\bullet j} &= \varepsilon \text{ для } j \in \{k | S_k \text{ не початково маркована позиція}\}, \\ (\mathbf{A}_0^* \mathbf{A}_1)_{\bullet j} &\neq \varepsilon \text{ для } j \in \{k | S_k \text{ початково маркована позиція}\}. \end{aligned}$$

В цьому випадку можна стверджувати, що початково марковані позиції графа синхронізації завжди виступають як вершини критичних циклів графа  $G(\mathbf{M})$ . Також з цього випливає можливість застосування власного числа, власних векторів і частотної проекції матриці  $\mathbf{M}$  для аналізу динамічних властивостей ДБС.

Поведінку в часі некерованого графа синхронізації можна дослідити за допомогою рівняння  $\mathbf{x}(k+1) = \mathbf{M} \mathbf{x}(k)$ , якщо задані початкові умови  $\mathbf{x}(0)$ . З урахуванням того, що циклічність критичного графа  $G^C(\mathbf{M})$  дорівнює 1, існує число  $K$ , таке що:

$$\forall k \geq K : \mathbf{M}^k = \lambda^k \mathbf{Q},$$

звідки випливає:

$$\mathbf{x}(k) = \lambda^k \mathbf{Q} \mathbf{x}(0) = \lambda^k \mathbf{v},$$

- де  $\lambda$  – власне число матриці  $\mathbf{M}$ ;  
 $\mathbf{v}$  – власний вектор матриці  $\mathbf{M}$ ;  
 $\mathbf{Q}$  – частотна проекція матриці  $\mathbf{M}$ .

На основі цього співвідношення можна стверджувати, що поведінка графа синхронізації для будь-яких початкових умов визначається сталим станом через деяку кількість пробігів по циклу в залежності від власного вектора  $\mathbf{v}$ .

В загальному випадку:

$$\forall k \geq K : \mathbf{M}^{k-\rho} = \lambda^\rho \mathbf{M}^k,$$

$$\mathbf{x}(k + \rho) = \mathbf{M}^{k+\rho} \mathbf{x}(0) = \lambda^\rho \mathbf{x}(k),$$

тобто в залежності від циклічності  $\rho$  графа  $G^C(\mathbf{M})$  заняття відповідних позицій графа синхронно повторюється після  $\lambda^\rho$  одиниць часу.

Прийнято вважати [59], що для  $k < K$  відповідні позиції займаються в нерегулярні часові проміжки і цей період може розглядатись як перехідний процес в системі.

### 8.3 Практична методика аналізу динаміки ДБС за допомогою “Max-plus” алгебри

Приклад, що ілюструє наведені вище твердження, розглядається для мережі Петрі, яка представлена на рис. 8.3 [37]. Програми обчислень наведені у додатку, лістинг 5.

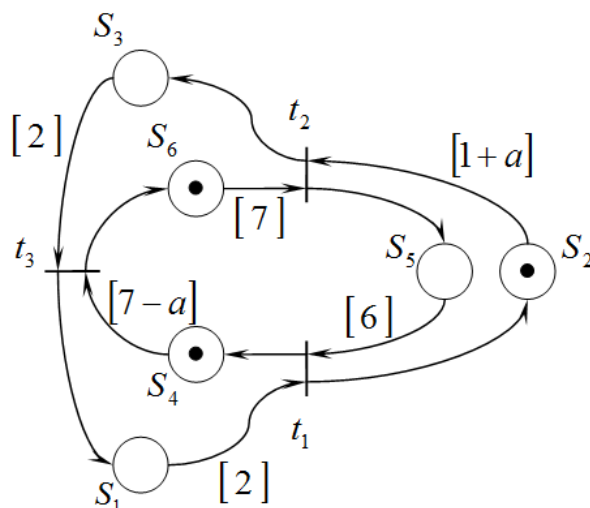


Рисунок 8.3 – Граф синхронізації мережі Петрі для абстрактного прикладу

ДБС має шість операційних позицій  $S_1 \dots S_6$ . Три з яких  $S_2, S_4, S_6$  – є початково маркованими.



Спочатку часовий параметр затримки  $a$  покладаємо рівним 0. В цьому випадку матриця динаміки  $\mathbf{A}$  матиме вигляд:

$$\mathbf{A} = \begin{bmatrix} \varepsilon & \varepsilon & 2 & 7 & \varepsilon & \varepsilon \\ 2 & \varepsilon & \varepsilon & \varepsilon & 6 & \varepsilon \\ \varepsilon & 1 & \varepsilon & \varepsilon & \varepsilon & 7 \\ 2 & \varepsilon & \varepsilon & \varepsilon & 6 & \varepsilon \\ \varepsilon & 1 & \varepsilon & \varepsilon & \varepsilon & 7 \\ \varepsilon & \varepsilon & 2 & 7 & \varepsilon & \varepsilon \end{bmatrix}.$$

За допомогою трансформуючих матриць і рівнянь (8.10) і (8.11) можуть бути розраховані матриці  $\mathbf{A}_0$  і  $\mathbf{A}_1$ :

$$\mathbf{A}_0 = \begin{bmatrix} \varepsilon & \varepsilon & 2 & \varepsilon & \varepsilon & \varepsilon \\ 2 & \varepsilon & \varepsilon & \varepsilon & 6 & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 2 & \varepsilon & \varepsilon & \varepsilon & 6 & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & 2 & \varepsilon & \varepsilon & \varepsilon \end{bmatrix},$$

$$\mathbf{A}_1 = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & 7 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & 1 & \varepsilon & \varepsilon & \varepsilon & 7 \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & 1 & \varepsilon & \varepsilon & \varepsilon & 7 \\ \varepsilon & \varepsilon & \varepsilon & 7 & \varepsilon & \varepsilon \end{bmatrix}.$$

З урахуванням відсутності додаткових логічних умов переключення немає необхідності в матриці  $\mathbf{B}$ .

З урахуванням (8.17) знаходимо  $\mathbf{A}_0^*$  і  $\mathbf{M}$ :

$$\mathbf{A}_0^* = \begin{bmatrix} 0 & \varepsilon & 2 & \varepsilon & \varepsilon & \varepsilon \\ 2 & 0 & 4 & \varepsilon & 6 & \varepsilon \\ \varepsilon & \varepsilon & 0 & \varepsilon & \varepsilon & \varepsilon \\ 2 & \varepsilon & 4 & 0 & 6 & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & 0 & \varepsilon \\ \varepsilon & \varepsilon & 2 & \varepsilon & \varepsilon & 0 \end{bmatrix},$$

$$\mathbf{M} = \begin{bmatrix} \varepsilon & 3 & \varepsilon & 7 & \varepsilon & 9 \\ \varepsilon & 7 & \varepsilon & 9 & \varepsilon & 13 \\ \varepsilon & 1 & \varepsilon & \varepsilon & \varepsilon & 7 \\ \varepsilon & 7 & \varepsilon & 9 & \varepsilon & 13 \\ \varepsilon & 1 & \varepsilon & \varepsilon & \varepsilon & 7 \\ \varepsilon & 3 & \varepsilon & 7 & \varepsilon & 9 \end{bmatrix}.$$

Структура графа  $G(\mathbf{M})$  матиме вигляд:

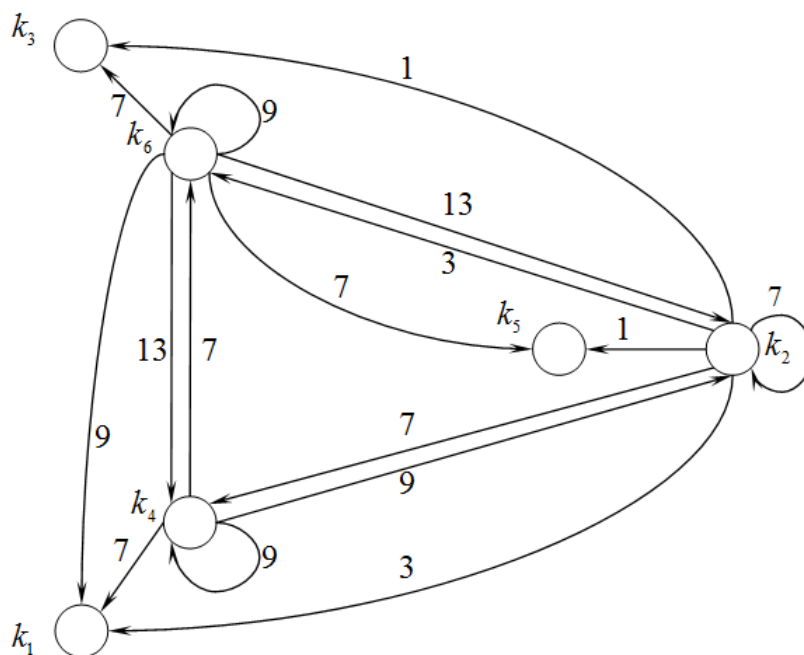


Рисунок 8.4 – Граф  $G(\mathbf{M})$

Сильно зв'язані вершини графа  $G(\mathbf{M})$  представлені на рис. 8.5.

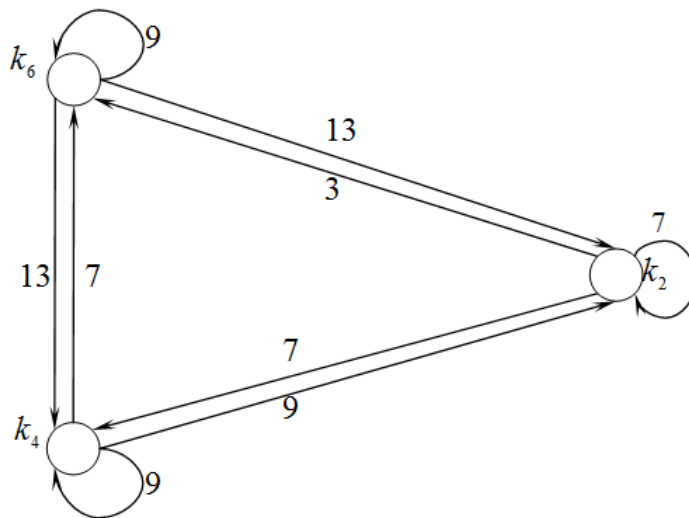


Рисунок 8.5 – Сильно зв’язані вершини графа  $G(\mathbf{M})$

Критичний граф  $G^C(\mathbf{M})$  представлено на рис. 8.6.

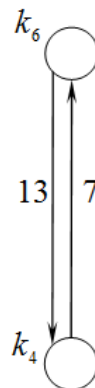


Рисунок 8.6 – Критичний граф  $G^C(\mathbf{M})$

Середня вага критичного циклу є власне число матриці  $\mathbf{M}$  –  $\lambda = (13 + 7) / 2 = 10$  одиниць часу.

Можна перевірити достовірність цього результату і за допомогою будь-якого з алгоритмів формального знаходження  $\lambda$  для матриці  $\mathbf{M}$ . Для власного числа матриці  $\mathbf{M}$  існує власний вектор:

$$\mathbf{v} = [3 \ 6 \ 0 \ 6 \ 0 \ 3]^T,$$

циклічність графа  $G^C(\mathbf{M})$   $\rho = 2$ , згідно з рис. 8.6.

При цьому перехідний процес закінчується власним станом ДБС тільки для деяких початкових умов.

Для прикладу розглянемо поведінку системи, якщо  $\mathbf{x}(0) = [0 \ 0 \ 0 \ 0 \ 0 \ 0]^T$ . А початково марковані позиції графа синхронізації відповідно візьмемо такими (рис. 8.3):

$$S_m = [0 \ 1 \ 0 \ 1 \ 0 \ 1].$$

Узагальнена характеристика динаміки ДБС по відношенню до позицій графа синхронізації  $S_i$  і змінних стану  $x_i$  представляється у вигляді індикативного вектора:

$$[0 \ 1(0) \ 0 \ 1(0) \ 0 \ 1(0)].$$

Обчислений за рівнянням (8.18), час маркування позицій  $S_i$  ( $i = \overline{1,6}$ ) у  $k$ -му циклі зводимо до таблиці:

$k$	0	1	2	3	...
$S_1$	0	9	20	29	...
$S_2$	0	13	22	33	...
$S_3$	0	7	16	27	...
$S_4$	0	13	22	33	...
$S_5$	0	7	16	27	...
$S_6$	0	9	20	29	...

Різниця між часом маркування для кожної з позицій у  $k$ -у і  $(k-1)$ -у циклах складає:

$k$	0	1	2	3	...
$S_1$	0	9	11	9	...
$S_2$	0	13	9	11	...
$S_3$	0	7	9	11	...
$S_4$	0	13	9	11	...
$S_5$	0	7	9	11	...
$S_6$	0	9	11	9	...

(8.19)

Позначимо час, що займає перехід  $t_j$  ( $j = \overline{1,3}$ ) у  $k$ -у циклі, через  $t_j^{(k)}$ . Розглянемо роботу першого циклу. З рис. 8.3 видно, що в результаті переходу  $t_2$  отримають мітки позиції  $S_3$  і  $S_5$ , а з (8.19) витікає, що це відбудеться через 7 одиниць часу. Отже, час переходу  $t_2$  дорівнює

$$t_2^{(1)} = 7,$$

а марковані позиції:

$$S = [0 \ 0 \ 1 \ 1 \ 1 \ 0].$$

Після роботи переходу  $t_3$  отримають мітки позиції  $S_1$  і  $S_6$ , час маркування яких складає 9 одиниць:

$$t_2^{(1)} + t_3^{(1)} = 9,$$

звідки

$$t_3^{(1)} = 9 - 7 = 2,$$

$$S = [1 \ 0 \ 0 \ 0 \ 1 \ 1].$$

Наступним відбувається перехід  $t_1$ , в результаті якого вдруге отримають мітки початково марковані позиції  $S_2$  і  $S_4$ . Проміжок часу між маркуванням складає 13 одиниць. Отримаємо:

$$t_2^{(1)} + t_3^{(1)} + t_1^{(1)} = 13,$$

отже,

$$t_1^{(1)} = 13 - 9 = 4$$

$$S = [0 \ 1 \ 0 \ 1 \ 0 \ 1].$$

Тобто, завершено перший цикл роботи системи.

На наступному циклі роботи системи позиції  $S_3$  і  $S_5$  отримують мітки через 9 одиниць часу від моменту першого маркування та включення переходів відповідно  $t_3$ ,  $t_1$  у першому циклі і  $t_2$  у другому циклі:

$$t_3^{(1)} + t_1^{(1)} + t_2^{(2)} = 9,$$

звідки

$$t_2^{(2)} = 9 - 2 - 4 = 3$$

і т.д.

Динамічна поведінка системи може бути представлена табулеграмою, яка наведена на рис. 8.7.

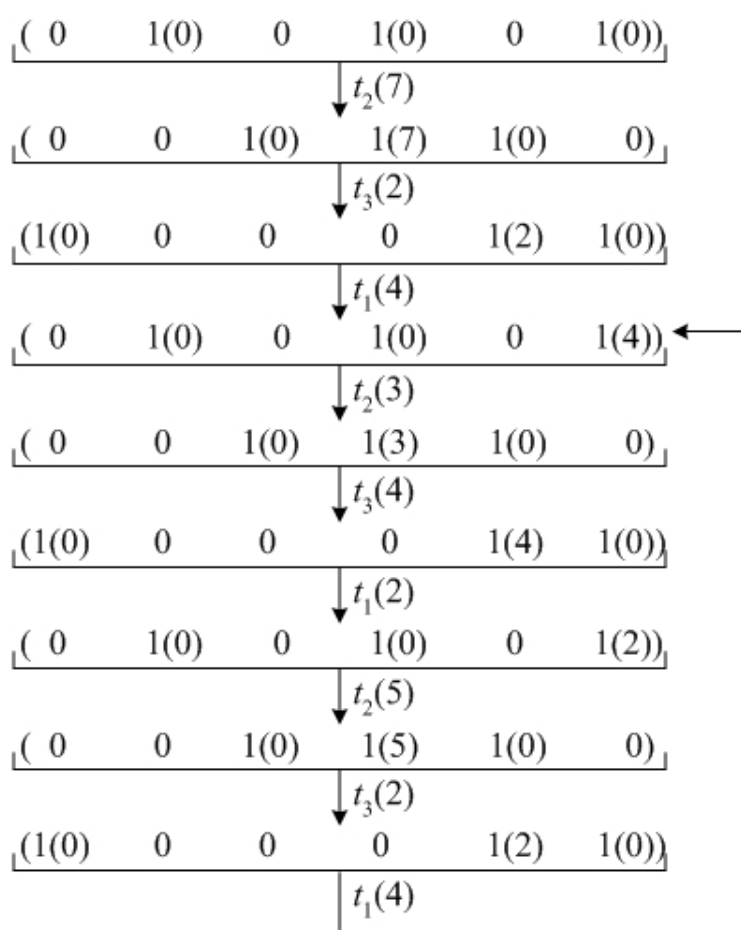


Рисунок 8.7 – Табулеграма динаміки ДБС

Очевидно, що в системі існує перехідний режим, який в подальшому призводить до встановлення сталого стану, який відображається повторенням індикативного вектора. Для того щоб уникнути перехідного режиму в системі необхідно в якості початкового вектора взяти власний вектор матриці  $\mathbf{M}$ , залишивши незмінним початкові позиції маркування графа синхронізації.

Якщо в якості початкового вектора взяти:

$$[0 \ 1(0) \ 0 \ 1(0) \ 0 \ 1(2)],$$

табулеграма поведінки ДБС в цьому випадку має вигляд представлений на рис. 8.8.

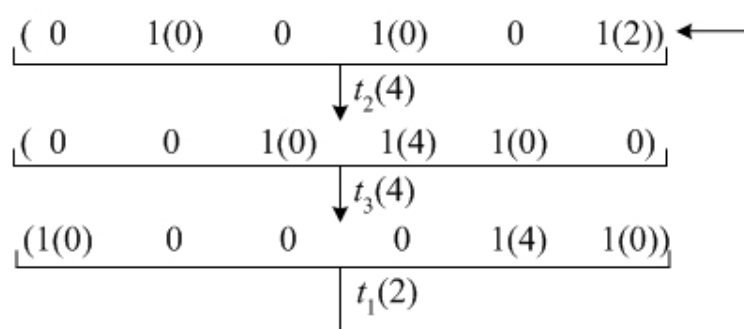


Рисунок 8.8 – Табулеграма поведінки ДБС для  $\mathbf{x}(0) = \mathbf{v} = [0 \ 0 \ 0 \ 0 \ 0 \ 2]^T$

З рисунку видно, що перехідний режим в системі відсутній і часовий проміжок між маркуванням позицій точно дорівнює власному числу матриці  $\mathbf{M}$ .

Поведінку графа синхронізації ДБС також можна більш детально пояснити за допомогою графіків, представлених на рис. 8.9.

З рис. 8.9 видно, що проміжок часу між  $k$ -м і  $(k + 1)$ -м маркуваннями в період перехідного процесу може бути більшим ніж власне число  $\lambda$  матриці  $\mathbf{M}$ . Після закінчення перехідного режиму маркування позиції відбувається по чергово через 9 і 11 одиниць часу. Середнє значення цього часового проміжку в точності дорівнює власному числу матриці  $\mathbf{M}$ .

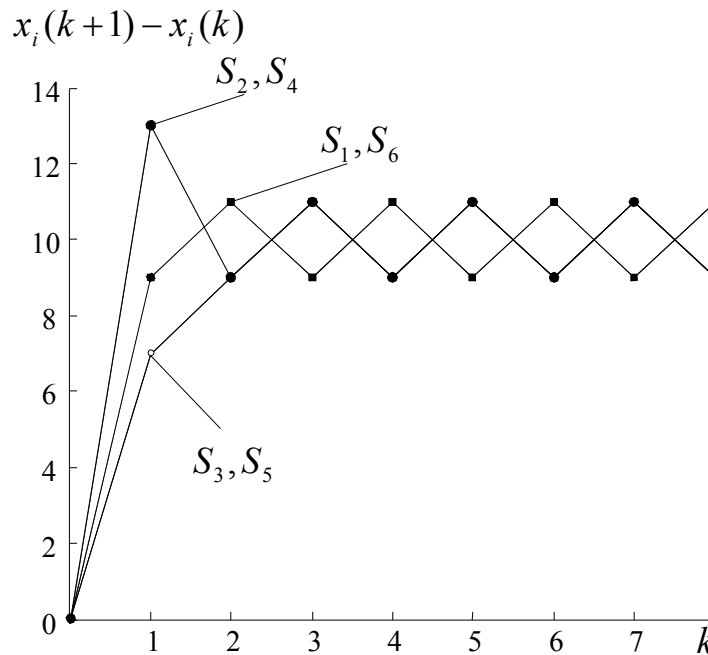


Рисунок 8.9 – Перехідний процес в ДБС при циклічності  $\rho = 2$

Якщо прийняти, що параметр  $a = 2$  в графі на рис. 8.3, то одержимо нову матрицю:

$$\mathbf{M}' = \begin{bmatrix} \varepsilon & 5 & \varepsilon & 5 & \varepsilon & 9 \\ \varepsilon & 9 & \varepsilon & 7 & \varepsilon & 13 \\ \varepsilon & 3 & \varepsilon & \varepsilon & \varepsilon & 7 \\ \varepsilon & 9 & \varepsilon & 7 & \varepsilon & 13 \\ \varepsilon & 3 & \varepsilon & \varepsilon & \varepsilon & 7 \\ \varepsilon & 5 & \varepsilon & 5 & \varepsilon & 9 \end{bmatrix}$$

з власним числом  $\lambda = 9$  і власним вектором  $\mathbf{v} = [2 \ 6 \ 0 \ 6 \ 0 \ 2]^T$ . Крім того циклічність графа зміниться на  $\rho = 1$ .

В цьому випадку будь-який вектор початкових умов  $\mathbf{x}(0)$  приводить систему у власний стан, при якому часовий проміжок маркування позицій точно дорівнює 9 одиниць часу, що відповідає власному числу матриці  $\mathbf{M}'$ . Табулеграма динаміки ДБС матиме такий вигляд:



*Представлення графа синхронізації ДБС в термінах мережі Петрі за допомогою “Max-Plus” алгебри*

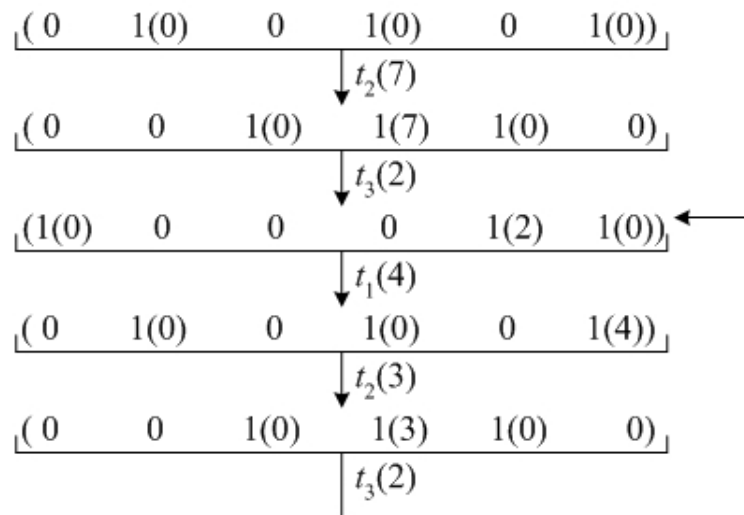


Рисунок 8.10 – Табулеграма динаміки ДБС при  $a = 2$ .

Графічна інтерпретація поведінки ДБС в динаміці проілюстрована на рис. 8.11.

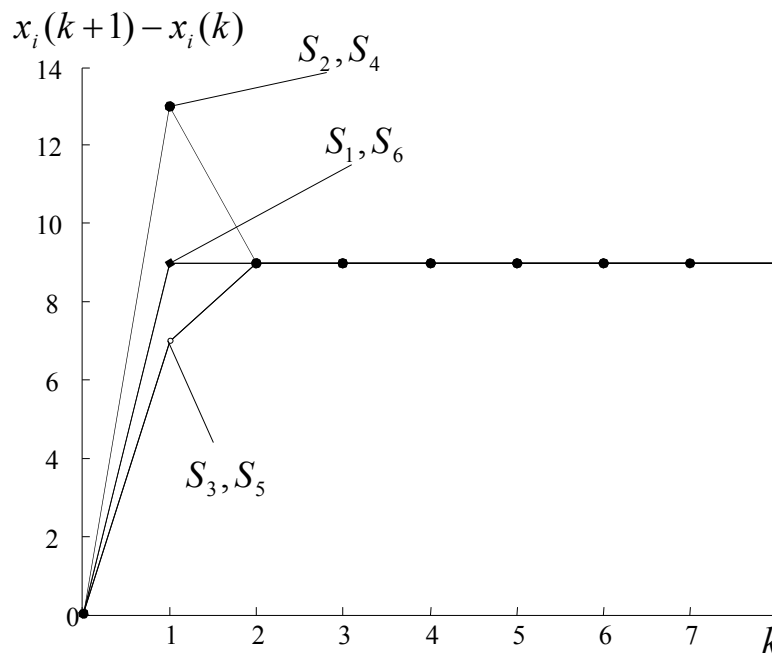


Рисунок 8.11 – Перехідний процес в ДБС при циклічності  $\rho = 1$

В продовж часу між  $k$  і  $(k+1)$  маркуваннями часовий відрізок може змінюватись (коливатись). Ця поведінка може трактуватись як перехідний процес в ДБС.

Якщо має місце  $k \geq K$ , то позиції маркуються через регулярні проміжки часу. Тобто, виникає послідовність маркувань, що повторюється в залежності від циклічності графа  $G(\mathbf{M})$ . Якщо для критичного графа  $G^c(\mathbf{M})$  циклічність  $\rho = 1$ , то проміжок часу між  $k$  і  $k+1$  маркуваннями складає точно  $\lambda$  одиниць часу. Поведінка ДБС на цьому етапі може розглядатись як власний усталений стан, або коротко – власний стан ДБС.

## 9 Основи синтезу управління в ДБС

Основою для розглянутого в цьому розділі синтезу управління є Max-Plus алгебраїчне представлення графа синхронізації згідно з рівнянням (8.18). Окремі елементи цієї методики представлено в публікаціях [58, 59]. Ідея синтезу управління в ДБС полягає в формуванні такого вектора управління, який задає дискретно-подійному процесу бажану поведінку. Задана поведінка ДБС може характеризуватись через множину недопустимих маркувань позицій або, якщо говорити про простір стану, заборонених станів системи. Тобто, деяка позиція не може маркуватись або деякий стан системи не може досягатись в конкретний момент часу.

Таке завдання поведінки ДБС означає, що шляхом управління необхідно затримати розвиток циклічного процесу в попередній позиції, щоб виключити можливість досягнення забороненого стану системи у відповідний момент часу. В загальному випадку правило заборони маркування позиції або досягнення відповідного стану формулюється наступним чином:

- Позиція  $S_i$  повинна маркуватись не раніше ніж через  $\tau$  одиниць часу після заняття маркером позиції  $S_j$ .

Таким чином, в ДБС задається послідовність заняття позицій, що рівноцінно по відношенню до графа синхронізації запобіганню заборонених станів.

### 9.1 Загальний підхід до синтезу управління

В розділі 8 показано, що динамічні властивості ДБС залежать від початкових умов  $x(0)$  вектора стану і власного вектора  $v$  матриці динаміки ДБС.

Тобто відповідним вибором вектора початкових умов та його співставленням з власним вектором можна досягти виключення заборонених станів системи. Але таке рішення задачі синтезу управління ДБС є досить примітивним і може бути реалізовано тільки в деяких специфічних системах.

Більш привабливою є ідея застосування керованих переходів з одного стану ДБС в інший. Для досягнення бажаної поведінки використовуються додаткові логічні умови переключення окремих переходів графа синхронізації. Часові точки маркувань (переключень) з урахуванням логічних умов сповільнюють в цілому розвиток процесу по відношенню до динаміки в системі з некерованими переходами.

Такий підхід дозволяє розглядати управління як результат дії зворотного зв'язку, за допомогою якого шляхом вирішення системи рівнянь Max-Plus алгебри можна формувати умови часових точок маркувань. Фактично за допомогою зворотного зв'язку і специфікації математичної моделі процесу можна говорити про класичну структуру системи управління в ДБС. При такій побудові управління в системі цілеспрямовано може змінюватись власне число і власний вектор в залежності від зовнішніх збурень, які безумовно діють на реальну ДБС.

Типова структура системи управління дискретно-безперервним об'єктом (ДБО) має вигляд, представлений на рис. 9.1.

Задача синтезу управління полягає в знаходженні матриць  $\mathbf{K}_0$  і  $\mathbf{K}_1$  з урахуванням заданої множини заборонених станів системи і заданої послідовності маркувань позиції графа синхронізації в часі. Очевидно, що в залежності від структури ДБО не завжди бажаний розвиток подій може бути реалізований в системі управління. Тому, для ДБС, як і для класичних автоматичних систем, характерним є поняття керованості. Але поняття керованості таких систем відрізняється від тотожного поняття для

класичних лінійних систем. Зокрема, на керованість ДБС суттєво впливають логічні умови керованих переходів, що не враховується в класичній теорії управління. Тому, проблема керованості в ДБС має специфічну інтерпретацію, потребує більш детального дослідження і в рамках цього розділу не розглядається.

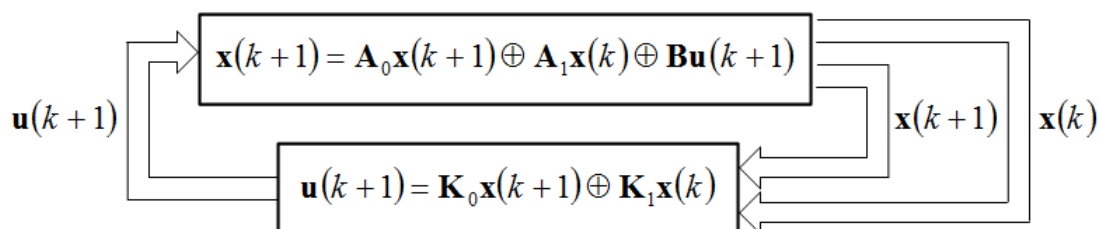


Рисунок 9.1 – Структура системи керування ДБО

В основі синтезу алгоритму управління лежить наступна ідея: оскільки шляхом управління розвиток у часі дискретно-безперервного процесу, залежного від власного числа матриці динаміки, може тільки сповільнюватись, то очевидною є потреба зменшення різниці між власними числами матриць керованої і некерованої систем. Тому, в якості критерію синтезу алгоритму управління може використовуватись оцінка різниці власних чисел некерованого процесу і процесу з керованими переходами.

Для того, щоб вплив завад був по можливості незначним, існує необхідність в дослідженні циклічності і кількості власних векторів в матриці динаміки ДБО.

Циклічність критичного графа  $G^C$  матриці динаміки керованої системи повинна дорівнювати  $\rho = 1$ . Це забезпечує в перехідному режимі встановлення стійкого циклу, який характеризується власним вектором матриці динаміки. Бажано, щоб сталі значення було однозначним. Це

можливо, коли в керованій системі матриці динаміки існує тільки один власний вектор.

Також, при вирішенні задачі синтезу алгоритму управління необхідно враховувати початкове маркування позицій в графі синхронізації, тому що Max-Plus алгебраїчна модель залежить від початкового розподілу маркувань.

Часові точки положень маркерів графа синхронізації задаються через вектор стану  $\mathbf{x}$  в рівнянні, яке детально розглядалось в розділі 8.

$$\mathbf{x}(k+1) = \mathbf{A}_0 \mathbf{x}(k+1) \oplus \mathbf{A}_1 \mathbf{x}(k) \oplus \mathbf{B} \mathbf{u}(k+1) \quad (9.1)$$

Структура системи зі зворотним зв'язком має загальний вигляд, як на рисунку 9.1 і описується рівнянням:

$$\mathbf{u}(k+1) = \mathbf{K}_0 \mathbf{x}(k+1) \oplus \mathbf{K}_1 \mathbf{x}(k) \quad (9.2)$$

При цьому матриці  $\mathbf{K}_0$  і  $\mathbf{K}_1$  – дві позитивно визначені  $(P \times |S|)$  матриці, які формують характер закону управління процесом. В цьому випадку значення  $P$  дорівнює числу логічних умов переключення керованим процесом,  $|S|$  – число вершин графа синхронізації. В результаті для керованого процесу маємо наступне рівняння в просторі стану замкнутої системи:

$$\mathbf{x}(k+1) = (\mathbf{A}_0 \oplus \mathbf{B} \mathbf{K}_0) \mathbf{x}(k+1) \oplus (\mathbf{A}_1 \oplus \mathbf{B} \mathbf{K}_1) \mathbf{x}(k). \quad (9.3)$$

Для того, щоб отримати розв'язок рівняння (9.3), а також розрахувати власне число і власний вектор, граф матриці  $\mathbf{A}_0 \oplus \mathbf{B} \mathbf{K}_0$  має бути вільним від циклів. За таких умов можливо визначення матриці  $\mathbf{M}_s$  керованої дискретно безперервної системи згідно з рівнянням:

$$\begin{aligned}
 \mathbf{x}(k+1) &= (\mathbf{A}_0 \oplus \mathbf{BK}_0)\mathbf{x}(k+1) \oplus (\mathbf{A}_1 \oplus \mathbf{BK}_1)\mathbf{x}(k) = \\
 &= (\mathbf{A}_0 \oplus \mathbf{BK}_0)^*(\mathbf{A}_1 \oplus \mathbf{BK}_1)\mathbf{x}(k) = \\
 &= \mathbf{M}_s \mathbf{x}(k).
 \end{aligned}
 \tag{9.4}$$

Описувана цим рівнянням поведінка графа синхронізації отримана з рівняння (8.18) за умови, що послідовність векторів управління  $\mathbf{u}(k)$ ,  $k = 0, 1, \dots$  є детермінованою. Цей випадок є особливим випадком поведінки системи і тому матриця  $\mathbf{M}_s$  може бути досліджена методами Max-Plus алгебри.

Рівняння (9.2) встановлює зв'язок між часовими точками маркування позицій  $k$ -о і  $(k+1)$ -о кроків управління. Для того щоб уникнути заборонених маркувань, які можуть з'явитися в поточному або наступних циклах управління, можна ввести додаткову умову, яка може бути врахована безпосередньо при розв'язанні рівняння (9.2).

Якщо є необхідність в завданні затримок на часові моменти маркування позицій на різних циклах управління, то з простих поточних умов може бути задіяна, як відповідне обмеження, найбільша поточна часова затримка.

Необхідною умовою синтезу управління в ДБС також є умова, що початкове маркування в системі відповідає допустимому стану об'єкта управління.

Простими вимогами до поведінки керованого графа синхронізації є наступні логічні умови:

1. Часова точка  $x_i(k)$   $k$ -о маркування позиції  $S_i$  має відбутися не раніше ніж через  $\tau$  одиниць часу після  $k$ -о маркування позиції  $S_j$ , що відповідає стану  $x_j(k)$ .

2. Часова точка  $x_i(k+1)$  ( $k+1$ )-о маркування позиції  $S_i$  має відбутися не раніше ніж через  $\tau$  одиниць часу після  $k$ -о маркування позиції  $S_j$  (стану  $x_j(k)$  попереднього  $k$ -о циклу маркування).

В першому випадку має місце умова затримки на заняття наступної позиції  $S_i$  при проходженні поточного циклу. Для виконання цієї умови необхідно знайти керований перехід із передобласті стану  $S_i$  мережі Петрі. В загальному випадку цей шлях може не включати початково маркованих положень графа синхронізації і не проходити через керовані переходи. Вага цього шляху дорівнює  $\delta$ , тобто не раніше ніж через  $\delta$  часових одиниць після включення переходу позицію  $S_i$  буде промарковано. Якщо існує декілька таких переходів, то вибирається шлях з максимальною вагою. Але якщо на перехід впливає управління  $u_m$ , то умова буде виконана через  $(\mathbf{K}_0)_{mj}$  одиниць часу, що дорівнює:

$$(\mathbf{K}_0)_{mj} = \tau \otimes (-\delta). \quad (9.5)$$

Це рівняння в залежності від  $x_j(k)$  встановлює найбільш ранню часову точку включення переходу. В ДБС завжди повинно виконуватися обмеження:

$$\tau \geq \delta. \quad (9.6)$$

Якщо це обмеження не виконується, то порушується послідовність подій в об'єкті, що є неприпустимим. При необхідності логічна умова може бути збережена шляхом збільшення  $\tau$ , але це не завжди можливо.

В описаному алгоритмі необхідно враховувати наступне:

Не завжди обмеження (9.6) може бути виконано, оскільки граф  $G(\mathbf{A}_0 \oplus \mathbf{BK}_0)$  має залишатись вільним від циклів, що досліджується в подальшому. Далі має бути визначено керований перехід. Також необхідно врахувати, що такий перехід може бути не знайдено на шляху від одного



стану до другого стану, який відповідає початково маркованій позиції. В цьому випадку маркування повинно відбуватись без впливу управління на переключення в бажаній часовій точці.

Аналогічно для іншого типу логічних вимог може бути знайдено елемент матриці  $\mathbf{K}_1$  :

$$(\mathbf{K}_1)_{mj} = \tau \otimes (-\delta), \quad (9.7)$$

який також задає системі бажану поведінку.

Для кожної окремої логічної вимоги може бути визначено елемент матриці зворотного зв'язку. Всі інші елементи, не пов'язані з додатковими логічними умовами і в обох матрицях  $\mathbf{K}_0$  і  $\mathbf{K}_1$ , дорівнюють  $\varepsilon$ .

Необхідно враховувати, що граф  $G(\mathbf{A}_0 \oplus \mathbf{BK}_0)$  повинен залишатись вільним від циклів (показник циклічності  $\rho \equiv 1$ ). Також не можна додавати додаткові шляхи в графі  $G(\mathbf{A}_0)$  в цикли, які проходять через ребра графа  $G(\mathbf{BK}_0)$ , тобто обумовлені наявністю додаткових логічних вимог. Таким чином, для того, щоб виконати ці обмеження можна застосувати ітераційну процедуру приведену в [57]:

1. В якості початкового значення задається  $\mathbf{K}_0 = \mathbf{N}$ ;
2. Якщо існує хоча б одна логічна умова, то це означає, що існує  $(\mathbf{K}_0)_{mj} \neq \varepsilon$ . Для його знаходження будується допоміжна матриця  $\mathbf{K}_H$  розмірності  $P \times |S|$ , в якій всі елементи  $(\mathbf{K}_H)_{mj} = (\mathbf{K}_0)_{mj} = \varepsilon$ . Тоді граф  $G(\mathbf{BK}_H)$  має ребра, які відповідають додатковим логічним умовам, і які всі виходять з вузлів  $(\mathbf{K}_H)_{j\bullet}$ . Тому, існує тільки один вектор-стовпець  $(\mathbf{BK}_H)_{\bullet j}$ , який має всі елементи, що дорівнюють  $\varepsilon$ .

3. В випадку, коли  $(\mathbf{BK}_H)_{ij} \neq \varepsilon$  і  $(\mathbf{A}_0 \oplus \mathbf{BK}_0)_{ji} \neq \varepsilon$ , додаткова логічна умова в матриці  $\mathbf{K}_0$  застосуватись не може. Тобто виконання цієї умови керування в ДБС не можливо.

4. Кроки 2 і 3 послідовно виконуються для всіх логічних умов, що задаються в системі, причому умовно третій крок виконується для логічних умов, які не виконувались раніше.

Очевидно, що цей ітераційний алгоритм залежить від послідовності застосування окремих логічних умов і має найкращу наглядність, коли ребра починаються з початково маркованого стану та послідовно розглядаються логічні умови, що входять до циклу, тобто перевіряються логічні умови в послідовності їх виконання в системі.

Оскільки управління тільки сповільнює процес, можливо лише збільшення власного числа  $\lambda_s$  матриці  $\mathbf{M}_s$  керованої системи по відношенню до власного числа некерованої системи. Відповідні власні вектори  $v_{s,i}$  показують, що в керованій системі логічні умови виконуються як в перехідному, так і в сталому стані розвитку динамічного процесу.

Приклад синтезу управління з логічними умовами проілюстровано на дискретно-безперервній системі, яка представлена мережею Петрі на рис. 9.2.

Програми обчислень наведені у додатку, лістинг 6.

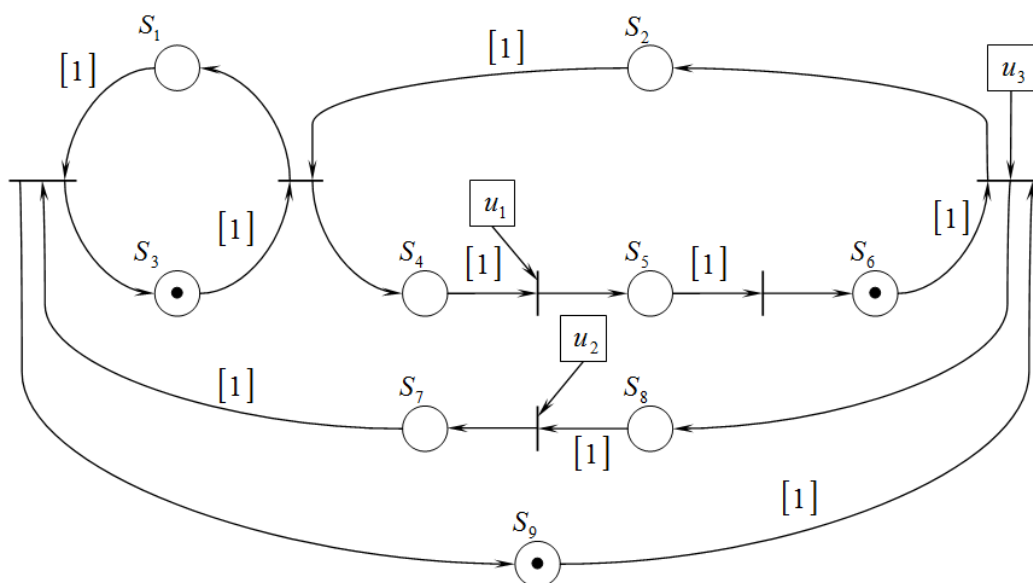


Рисунок 9.2 – Граф синхронізації керованої системи

Представляючи цю мережу в просторі станів, отримаємо такий вид матриці динаміки  $\mathbf{A}$  і матриці управління  $\mathbf{B}$ :

$$\mathbf{A} = \begin{bmatrix} \varepsilon & 1 & 1 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 1 & \varepsilon & \varepsilon & 1 \\ 1 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 1 & \varepsilon & \varepsilon \\ \varepsilon & 1 & 1 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & 1 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & 1 & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 1 & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 1 & \varepsilon & \varepsilon & 1 \\ 1 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 1 & \varepsilon & \varepsilon \end{bmatrix}, \mathbf{B} = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & 0 \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \\ 0 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & 0 & \varepsilon \\ \varepsilon & \varepsilon & 0 \\ \varepsilon & \varepsilon & \varepsilon \end{bmatrix}$$

Вектор початкового маркування позицій матиме наступний вигляд:

$$m_0 = [0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1]^T.$$

Матриці  $\mathbf{A}^*$  і  $\mathbf{M}$  досить легко обчислюються і мають наступний вигляд:

$$\mathbf{A}_0^* = \begin{bmatrix} 0 & 1 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & 0 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 1 & 2 & 0 & \varepsilon & \varepsilon & \varepsilon & 1 & 2 & \varepsilon \\ \varepsilon & 1 & \varepsilon & 0 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & 2 & \varepsilon & 1 & 0 & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & 3 & \varepsilon & 2 & 1 & 0 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 0 & 1 & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 0 & \varepsilon \\ 1 & 2 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 1 & 2 & 0 \end{bmatrix};$$

$$\mathbf{M} = \begin{bmatrix} \varepsilon & \varepsilon & 1 & \varepsilon & \varepsilon & 2 & \varepsilon & \varepsilon & 2 \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 1 & \varepsilon & \varepsilon & 1 \\ \varepsilon & \varepsilon & 2 & \varepsilon & \varepsilon & 3 & \varepsilon & \varepsilon & 3 \\ \varepsilon & \varepsilon & 1 & \varepsilon & \varepsilon & 2 & \varepsilon & \varepsilon & 2 \\ \varepsilon & \varepsilon & 2 & \varepsilon & \varepsilon & 3 & \varepsilon & \varepsilon & 3 \\ \varepsilon & \varepsilon & 3 & \varepsilon & \varepsilon & 4 & \varepsilon & \varepsilon & 4 \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 2 & \varepsilon & \varepsilon & 2 \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 1 & \varepsilon & \varepsilon & 1 \\ \varepsilon & \varepsilon & 2 & \varepsilon & \varepsilon & 3 & \varepsilon & \varepsilon & 3 \end{bmatrix}.$$

Матриця  $\mathbf{M}$  має власне число  $\lambda = 4$  і власний вектор:

$$\mathbf{v} = [1 \ 0 \ 2 \ 1 \ 2 \ 3 \ 1 \ 0 \ 2]^T.$$

Також в керованій системі мають виконуватись такі логічні умови:

1. Позиція  $S_6$  повинна маркуватись в поточному циклі через 2 часові одиниці після демаркування позиції  $S_3$ .
2. Позиція  $S_7$  повинна маркуватись в поточному циклі через одну часову одиницю після демаркування позиції  $S_5$ .
3. В наступному  $(k + 1)$  циклі позиція  $S_2$  повинна бути маркована через 3 одиниці часу після демаркування позиції  $S_9$  в попередньому циклі.

Логічні умови 1 і 2 можуть бути реалізовані шляхом внесення їх в матрицю  $\mathbf{K}_0$ . Зрозуміло, що необхідно перевірити за допомогою ітераційного алгоритму, чи з'являються додаткові цикли в  $G(\mathbf{A}_0 \oplus \mathbf{BK}_0)$ .

Перша умова може бути виконана за допомогою керованого переходу  $u_1$ . Цей перехід має в своїй післяобласті позицію  $S_5$ , з якої веде шлях вагою  $\delta_1 = 1$  до позиції  $S_6$ .

Визначимо допоміжну матрицю  $\mathbf{K}_H$ . Елемент цієї матриці  $(\mathbf{K}_H)_{13} = \tau_1 \otimes (-\delta_1) = 2 \otimes (-1) = 1$ , всі інші елементи матриці  $\mathbf{K}_H = \varepsilon$ . Тому  $(\mathbf{BK}_H)_{53} = 1$  і перша логічна умова може бути виконана при умові, що  $(\mathbf{K}_0)_{13} = 1$ , оскільки  $(\mathbf{A}_0^*)_{35} = \varepsilon$ .

Для другої логічної умови елемент  $(\mathbf{K}_H)_{25} = \tau_2 \otimes (-\delta_2) = 1 \otimes 0 = 1$ ,  $(\mathbf{BK}_H)_{75} = 1$ .

З урахуванням цієї логічної умови матриця  $(\mathbf{A}_0 \oplus \mathbf{BK}_0)^*$  має вигляд:

$$(\mathbf{A}_0 \oplus \mathbf{BK}_0)^* = \begin{bmatrix} 0 & 1 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & 0 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 1 & 2 & 0 & \varepsilon & \varepsilon & \varepsilon & 1 & 2 & \varepsilon \\ \varepsilon & 1 & \varepsilon & 0 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 2 & 3 & 1 & 1 & 0 & \varepsilon & 2 & 3 & \varepsilon \\ 3 & 4 & 2 & 2 & 1 & 0 & 3 & 4 & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 0 & 1 & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 0 & \varepsilon \\ 1 & 2 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 1 & 2 & 0 \end{bmatrix}.$$

Тобто, має місце факт, що  $(\mathbf{A}_0 \oplus \mathbf{BK}_0)_{57}^* = 2$ . Це говорить про те, що логічна умова 2 не може бути виконана.

Третя логічна умова виконується у разі, коли  $(\mathbf{K}_1)_{39} = 3$ . Тоді матриці зворотного зв'язку  $\mathbf{K}_0$  і  $\mathbf{K}_1$  мають наступний вигляд:

$$\mathbf{K}_0 = \begin{bmatrix} \varepsilon & \varepsilon & 1 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{bmatrix};$$

$$\mathbf{K}_1 = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 3 \end{bmatrix}.$$

Матриця замкнутої некерованої системи  $\mathbf{M}_s$  матиме вигляд:

$$\mathbf{M}_s = (\mathbf{A}_0 \oplus \mathbf{BK}_0)^* \cdot (\mathbf{A}_1 \oplus \mathbf{BK}_1) = \begin{bmatrix} \varepsilon & \varepsilon & 1 & \varepsilon & \varepsilon & 2 & \varepsilon & \varepsilon & 4 \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 1 & \varepsilon & \varepsilon & 3 \\ \varepsilon & \varepsilon & 2 & \varepsilon & \varepsilon & 3 & \varepsilon & \varepsilon & 5 \\ \varepsilon & \varepsilon & 1 & \varepsilon & \varepsilon & 2 & \varepsilon & \varepsilon & 4 \\ \varepsilon & \varepsilon & 3 & \varepsilon & \varepsilon & 4 & \varepsilon & \varepsilon & 6 \\ \varepsilon & \varepsilon & 4 & \varepsilon & \varepsilon & 5 & \varepsilon & \varepsilon & 7 \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 2 & \varepsilon & \varepsilon & 4 \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 1 & \varepsilon & \varepsilon & 3 \\ \varepsilon & \varepsilon & 2 & \varepsilon & \varepsilon & 3 & \varepsilon & \varepsilon & 5 \end{bmatrix}.$$

Власне число матриці  $\mathbf{M}_s$   $\lambda_s = 5$ . Таким чином поведінка керованого процесу на одну одиницю часу повільніша, ніж некерований процес. Результат управління можна оцінювати за допомогою вектора стану або власного вектора.

Таблиця 9.1 демонструє, як змінюється вектор стану для вибраної позиції графу синхронізації. Для наглядності і порівняння дані в таблиці представлені в “накладеному” вигляді.

Власний вектор керованої системи матиме вигляд:

$$\mathbf{v}_s = [1 \ 0 \ 2 \ 1 \ 2 \ 3 \ 1 \ 0 \ 2]^T$$

та ілюструє бажану поведінку замкнутої системи.

Таблиця 9.1 – Часові точки маркування позицій графа синхронізації

$S \backslash k$	0	1	2	3	4	5
$S_2$		1 3	5 8	9 13	13 18	17 23
$S_3$	0 0	3 5	7 10	11 15	15 20	19 25
$S_5$		3 6	7 11	11 16	15 21	19 26
$S_6$	0 0	4 7	8 12	12 17	16 22	20 27
$S_7$		2 4	6 9	10 14	14 19	18 24
$S_9$	0 0	3 5	7 10	11 15	15 20	19 25

Аналізуючи дані таблиці 9.1, можна бачити, що логічні умови 1 і 3 виконуються. Наприклад, в третьому циклі ( $k = 3$ ) позиція  $S_9$  займає часову точку 15. Позиція  $S_2$  в наступному циклі ( $k = 4$ ) маркується в часовій точці 18, і необхідний інтервал затримки (3 одиниці часу) відповідає третій логічній умові.

## 9.2 Структура і властивості матриці динаміки керованої ДБС

Структуру матриці  $\mathbf{M}_s$  можна проаналізувати на основі рівняння (9.4). В графі  $G(\mathbf{A}_0 \oplus \mathbf{BK}_0)$  завжди присутні вершини і дуги графа  $G(\mathbf{A}_0)$ . Також в графі  $G(\mathbf{A}_1 \oplus \mathbf{BK}_1)$  завжди присутні вершини і дуги графа  $G(\mathbf{A}_1)$ . Тому можна стверджувати, що в графі  $G(\mathbf{M}_s)$  кількість сильно зв'язаних вершин може тільки бути збільшено по відношенню до графа  $G(\mathbf{M})$ .

Якщо прийняти, що граф  $G(\mathbf{BK}_1)$  є визначеним, і вузол  $K_j$  в графі  $G(\mathbf{M}_s)$  має вихідні ребра, що відповідають післяобласті позиції  $S_j$  графа синхронізації. Також вимагається, що ця позиція повинна маркуватись при

виконанні логічної умови перед позицією  $S_i$ , причому ця умова враховується в матриці  $\mathbf{K}_1$ . Тоді можна стверджувати, що вузли графа матриці замкнутої системи  $\mathbf{M}_s$  належать деякій множині  $\mathbf{V}_{\mathbf{K}_1}$ :

$$\mathbf{V}_{\mathbf{K}_1} = \{K_j \in G(\mathbf{M}_s)\} \quad \exists m: (\mathbf{K}_1)_{mj} \neq \varepsilon. \quad (9.8)$$

Це твердження має строгий доказ, який приведено в [58]. Практична цінність цього доказу полягає в тому, що для всіх  $K_j \in G(\mathbf{V}_{\mathbf{K}_1})$  існує хоча б одне ребро від  $K_j$  до вершини графа  $G(\mathbf{M})$ , яке належить до сильно зв'язаних компонентів цього графа, а тому сильно зв'язані компоненти графа  $G(\mathbf{M}_s)$  завжди включають вузли  $K_j \in G(\mathbf{V}_{\mathbf{K}_1})$ . Також, як висновок із теореми, приведеної в [57], очевидно, що при старті керованого графа синхронізації виконується рівняння:

$$\mathbf{x}(1) = \mathbf{M}_s \cdot \mathbf{x}(0), \quad (9.9)$$

та повинні бути задані часові точки маркування, хоча б однієї початково немаркованої позиції. Це обумовлено структурою зворотного зв'язку, який формує найбільш ранні часові точки керованих переходів і часові точки маркувань початково не зайнятих позицій. Оскільки ці часові точки при старті не відомі, існує проблема стартового управління на першому кроці управління ДБС.

### **9.3 Аналіз проблеми керування на стартовому циклі**

Якщо всі логічні умови управління ДБС, що знайшли своє відображення в матриці зворотного зв'язку  $\mathbf{K}_1$ , сповільнюють розвиток процесу виключно відносно початково маркованого положення, то керований граф синхронізації може бути запущений шляхом задавання часових точок маркування початково маркованих позицій графа. В цьому



випадку сильно зв'язані компоненти графів  $G(\mathbf{M})$  і  $G(\mathbf{M}_s)$  є повністю ідентичними.

Якщо ця умова не виконується, тоді часові точки маркування позицій графа синхронізації повинні бути задані не для початково маркованих позицій. В цьому випадку для першого циклу переходів ДБС розроблюють спеціальне управління, яке має назву стартового управління. Математичний опис системи в першому циклі має наступний вигляд:

$$\mathbf{x}(1) = \mathbf{A}_0\mathbf{x}(1) \oplus \mathbf{A}_1\mathbf{x}(0) \oplus \mathbf{B}\mathbf{u}(1), \quad (9.10)$$

або для замкнутої системи:

$$\mathbf{x}(1) = \mathbf{M}\mathbf{x}(0) \oplus \mathbf{A}_0^*\mathbf{x}(0) \oplus \mathbf{B}\mathbf{u}(1). \quad (9.11)$$

Таким чином, до моменту старту керованого графа синхронізації поряд з часовими точками маркування початково зайнятих позицій повинно бути визначено час переключення  $\mathbf{u}(1)$  керованого переходу.

В попередньому розділі при розгляді динаміки ДБС управління на початкових позиціях не приймалося до уваги і вважалось, що стартове управління формується з власного вектора ДБС. Такий підхід виключає перехідний процес і система завжди знаходиться у власному стані. Також скорочується число станів системи і логічні умови керованих переходів завжди можуть бути виконані. Якщо при розгляді управління в ДБС абстрагуватись від структури зі зворотним зв'язком, то стартове управління з власним вектором дозволяє досягти режиму без перехідного процесу при будь-яких часових моментах маркування початково маркованих позицій.

Реальна ДБС, як правило, стартує з довільних початкових умов  $\mathbf{x}(0)$ , які не завжди співпадають з власним вектором  $\mathbf{v}$  об'єкта. Якщо початкова маркована позиція  $S_i$  в своїй післяобласті не має керованих переходів,

можна маркувати наступне положення в часовій точці, яка не відповідає власній поведінці. Але для однозначності прийнято задавати  $x_i(0) = v_i$  з тим, щоб наступні позиції маркувались відносно власного вектора. Ця умова повинна виконуватись для всіх  $i$ -х компонент власного вектора, які відповідають позиціям  $S_i$  графа синхронізації, які в своїй післяобласті не мають керованих переходів. Для початково маркованих позицій, які в своїй післяобласті мають керовані переходи, відповідна початкова часова точка повинна бути меншою, ніж відповідна компонента власного вектора.

Підсумовуючи, можна сформулювати наступні умови старту ДБС з власним вектором початкових умов:

1.  $v_i > x_i(0)$ , якщо позиція  $S_i$  є початково маркованою і знаходиться в передобласті керованого переходу;
2.  $v_i = x_i(0)$ , якщо позиція  $S_i$  є початково маркованою і знаходиться в передобласті некерованого переходу.

Зрозуміло, що старт з початкових умов, що відповідають власному вектору стану, залежить від структури мережі Петрі і початкових умов, які задаються для вектора стану  $\mathbf{x}(0)$ . В деяких об'єктах старт можливий тільки з визначених позицій, для інших – з будь-яких позицій, які пов'язані з власним вектором матриці динаміки системи. Але в будь-якому випадку всі початкові маркування в мережі Петрі повинні бути задані, а вектор початкових умов повинен бути сформований згідно з умовами старту ДБС, що наведені вище.

Стартовий вектор управління  $\mathbf{u}(1)$  при цьому може бути визначений як:

$$\mathbf{u}(1) = \mathbf{K}_0 \mathbf{x}(1) \oplus (\mathbf{B}^T \mathbf{A}_1 \oplus \mathbf{K}_1) \mathbf{v}_s, \quad (9.12)$$

і ДБС переводиться у власний стан:

$$\begin{aligned} \mathbf{x}(1) &= \mathbf{A}_0 \mathbf{x}(1) \oplus \mathbf{A}_1 \mathbf{x}(0) \oplus \mathbf{B} \mathbf{u}(1) = \\ &= (\mathbf{A}_0 \oplus \mathbf{B} \mathbf{K}_0)^* (\mathbf{A}_1 \mathbf{x}(0) \oplus (\mathbf{B} \mathbf{B}^T \mathbf{A}_1 \oplus \mathbf{B} \mathbf{K}_1) \mathbf{v}_s). \end{aligned} \quad (9.13)$$

Справедливість виразу (9.13) витікає з наступних міркувань. Елемент  $(\mathbf{A}_1)_{ij} \neq \varepsilon$  тільки тоді, коли позиція  $S_j$  лежить позаду початково маркованої позиції  $S_i$ . Згідно з вищенаведеними умовами старту ДБС з початковими умовами, що відповідають власному вектору системи, маємо наступне:

$$(\mathbf{A}_1 \mathbf{v}_s)_i \begin{cases} \geq (\mathbf{A}_1 \mathbf{x}(0))_i, & \text{якщо } S_i \text{ – позиція, що знаходиться позаду} \\ & \text{керованого переходу;} \\ = (\mathbf{A}_1 \mathbf{x}(0))_i, & \text{в іншому випадку.} \end{cases} \quad (9.14)$$

Окрім того має місце:

$$(\mathbf{B} \mathbf{B}^T)_{ij} = \begin{cases} 0, & \text{якщо позиції } S_i \text{ і } S_j \text{ займаються одночасно і} \\ & \text{знаходяться позаду керованого переходу;} \\ \varepsilon, & \text{в іншому випадку.} \end{cases} \quad (9.15)$$

Також

$$\begin{aligned} (\mathbf{B} \mathbf{B}^T \mathbf{A}_1 \mathbf{v}_s)_i &= \bigoplus_{j=1}^{|S|} (\mathbf{B} \mathbf{B}^T)_{ij} (\mathbf{A}_1 \mathbf{v}_s)_j = \\ &= \begin{cases} (\mathbf{A}_1 \mathbf{v}_s)_i, & \text{якщо позиція } S_i \text{ знаходиться позаду керованого} \\ & \text{переходу і початково маркованої позиції;} \\ \varepsilon, & \text{в іншому випадку.} \end{cases} \end{aligned} \quad (9.16)$$

Таким чином, з урахуванням (9.14) – (9.16) маємо:

$$(\mathbf{A}_1 \mathbf{x}(0) \oplus \mathbf{B} \mathbf{B}^T \mathbf{A}_1 \mathbf{v}_s)_i =$$

$$= \begin{cases} (\mathbf{A}_1 \mathbf{x}(0) \oplus \mathbf{A}_1 \mathbf{v}_s)_i = (\mathbf{A}_1 \mathbf{v}_s)_i, & \text{якщо позиція } S_i \text{ знаходиться позаду} \\ & \text{керованого переходу і початково} \\ & \text{маркованої позиції;} \\ (\mathbf{A}_1 \mathbf{x}(0))_i = (\mathbf{A}_1 \mathbf{v}_s)_i, & \text{якщо позиція } S_i \text{ знаходиться позаду} \\ & \text{некерованого переходу і початково} \\ & \text{маркованої позиції;} \\ \varepsilon = (\mathbf{A}_1 \mathbf{v}_s)_i, & \text{в іншому випадку.} \end{cases}$$

і остаточно:

$$(\mathbf{A}_1 \mathbf{x}(0) \oplus (\mathbf{B}\mathbf{B}^T \mathbf{A}_1 \oplus \mathbf{B}\mathbf{K}_1) \mathbf{v}_s) = (\mathbf{A}_1 \oplus \mathbf{B}\mathbf{K}_1) \mathbf{v}_s. \quad (9.17)$$

Є деякі особливості знаходження стартового управління, враховуючи рівняння (9.12). Вектор стану  $\mathbf{x}(1)$  на початку розвитку процесу є невідомим, а тому управління  $\mathbf{u}(1)$  також є не повністю визначеним. Але построкове рішення системи (9.12) дозволяє знаходити всі компоненти  $x_i(1)$  раніше, ніж відповідні компоненти  $u_i(1)$  вектора  $\mathbf{u}(1)$ . Також необхідно мати на увазі твердження із [58], а саме: якщо структура ДБС не дозволяє забезпечити виконання будь-якої з двох вимог до стартового управління, то систему можна перевести у власний стан, але при цьому рівняння для стартового управління необхідно змінити, як рекомендовано в [59], на:

$$\mathbf{u}(1) = \mathbf{K}_0 \mathbf{x}(1) \oplus (\mathbf{B}^T \mathbf{A}_1 \oplus \mathbf{K}_1) \mathbf{v}_s \oplus \lambda_s \mathbf{B}^T \mathbf{v}_s, \quad (9.18)$$

де додатковий член рівняння  $\lambda_s \mathbf{B}^T \mathbf{v}_s$  гарантує, що всі позиції, які знаходяться в післяобласті відносно керованого переходу, не займаються раніше ніж часові моменти, які відповідають власному вектору.

Шляхом відповідного масштабування компонент власного вектора можна досягти, щоб завжди виконувалась нерівність  $v_{s,i} \geq x_i(0)$ , яка гарантує, що в системі завжди відповідний перехід буде виконано раніше

ніж відповідна логічна умова керування почне діяти в стартовому управлінні. А це означає, що наступна за керованим переходом позиція вже маркується як у власному стані.

#### 9.4 Особливості реалізації управління в ДБС з жорсткими логічними умовами

Поведінка системи в часі цілком визначається власним вектором матриці динаміки системи  $\mathbf{M}_s$ , якщо граф синхронізації ДБС знаходиться у власному стані. Згідно з розглянутою теорією інваріантності, управління може бути сформовано логічними умовами, які включаються до матриць зворотного зв'язку  $\mathbf{K}_0$  або  $\mathbf{K}_1$ . Причому, у власному стані поведінки можна модифікувати зворотній зв'язок і залишити тільки одну матрицю. Але між цими варіантами модифікації існують істотні відмінності з точки зору формування управління. Так, якщо залишити  $\mathbf{K}_0$ , маємо рівняння:

$$u_i(k+1) = (\mathbf{K}_0)_{ij} x_j(k+1) = r \cdot x_j(k+1),$$

в той же час, якщо залишити тільки  $\mathbf{K}_1$ , маємо інше рівняння:

$$u_i(k+1) = (\mathbf{K}_1)_{ij} x_j(k) = \lambda_s \cdot r \cdot x_j(k).$$

В першому випадку розрахунок управління відбувається в поточному циклі системи. Якщо логічна умова включається до  $\mathbf{K}_1$ , то управління формується з часових моментів попереднього циклу. Оскільки у власному стані всі позиції через  $\lambda_s$  одиниць часу маркуються заново,  $(k+1)$ -е маркування може бути розраховане з  $k$ -о за допомогою власного числа матриці замкнутої системи.

В загальному випадку управління, сформоване з використанням  $\mathbf{K}_0$ , може реагувати на логічну умову більш чутливо, оскільки розраховується безпосередньо в поточному циклі. Разом з цим, можливість розрахунку

моментів маркувань між різними циклами, дозволяє враховувати більш складні логічні умови управління. Так, наприклад, якщо в деякій системі існують дві позиції, які маркуються у власному стані одночасно, то перше положення повинно маркуватись в  $k$ -у циклі, а друге в  $(k+1)$ -у. Це обмеження можна врахувати в логічних умовах управління і таке управління буде мати назву управління з жорсткими умовами.

Для реалізації управління з жорсткими умовами необхідним є старт об'єкта управління з початковими умовами, що відповідають власному вектору некерованої системи  $\mathbf{v}$ . За допомогою власного вектора об'єкта можна виявити всі заборонені стани, які є підмножиною заборонених станів замкнутої керованої системи.

В загальному випадку заборонений стан можна виключити шляхом застосування обмежень. Часова дистанція маркування розглядається як складна конструкція, що описується формулою:

$$x_i(k) = \tau_{ist} \otimes x_j(k + K_{ist}).$$

В цьому випадку  $\tau_{ist}$  є прямий власний шлях між маркуванням обох позицій, а  $K_{ist}$  вказує цикл, в якому ця подія відбувається. Тобто, для виключення зворотного зв'язку повинно виконуватись обмеження:

$$0 < \tau_{ist} < \lambda.$$

На основі власного стану та початкових умов, що відповідають власному вектору системи, маємо рівняння:

$$x_i(k) = \tau_{ist} \otimes \lambda^{K_{ist}} \otimes x_j(k).$$

Більш жорсткі логічні умови управління в часовому відрізку відносно  $\tau$  і  $k$  будуть реалізовані шляхом їх урахування в матриці зворотного зв'язку  $\mathbf{K}_0$ :

$$(\mathbf{K}_0)_{ij} = \delta^{-1} \otimes \tau \otimes \lambda_s^K.$$

Таким чином, за допомогою методики, що розглянута в підрозділі 9.1, може бути розрахована матриця зворотного зв'язку  $\mathbf{K}_0$ . Також для матриці динаміки замкнутої системи  $\mathbf{M}_s$  визначається власне число  $\lambda_s$ . При цьому необхідно враховувати, що всі елементи матриці  $\mathbf{K}_0$ , відмінні від  $\varepsilon$ , повинні бути більшими або рівними  $e$ , що забезпечує задану послідовність в графі синхронізації керованої системи.

Наприклад, якщо дві позиції  $S_1$  і  $S_2$  маркується одночасно, але частота маркувань відрізняється:

$$x_1(k) = x_2(k + K_{ist}),$$

то за допомогою логічних умов управління може бути затримано маркування позиції  $S_1$  по відношенню до  $S_2$  на  $\tau$  одиниць часу. Але не навпаки, оскільки, в цьому випадку при  $\delta = 0$  в матриці  $\mathbf{K}_0$  можна отримати від'ємний елемент, що є недопустимим:

$$(\mathbf{K}_0)_{mj} = \tau \otimes \lambda_s^{-K} < 0.$$

Власне число керованої ДБС  $\lambda_s$  завжди більше власного числа некерованої системи. Оскільки часові точки маркувань завжди сповільнюють процес переключень в порівнянні з розвитком процесу в часі для некерованої системи. Але в цьому випадку розрахунок власного значення матриці  $\mathbf{M}_s$  є досить складною чисельною задачею. В загальному випадку для знаходження власного числа керованої системи існує рівняння:

$$\mathbf{M}_s \mathbf{v}_s = ((\mathbf{A}_0 \oplus \mathbf{BK}_0(\lambda_s))^* (\mathbf{A}_1 \oplus \mathbf{BK}_1)) \mathbf{v}_s = \lambda_s \mathbf{v}_s. \quad (9.19)$$

Розглядаючи рівняння (9.19), необхідно дати відповідь на два питання:

1. За яких умов існує власне число  $\lambda_s$  і власний вектор  $\mathbf{v}_s$ , які дають розв'язок рівняння.

2. Чи є знайдений вектор  $\mathbf{v}_s$  рішенням, що відповідає синтезу управління з жорсткими умовами.

Математичний доказ відповіді на ці питання ґрунтується на наступних постулатах і твердженнях, які більш детально представлено в [57].

Кожна логічна умова управління в ДБС затримує маркування позиції  $S_i$  по відношенню до  $S_j$  на  $\tau$  одиниць часу, що пов'язано з керованим переходом  $t_s$ , на який впливає управління  $u_m$ . Із довільної позиції  $S_k$ , такої що  $S_k \in t_s$ , завжди існує елементарний шлях з вагою  $\delta$  до позиції  $S_i$ , причому для кожної позиції окрім  $S_i$ , має місце обмеження  $S_k \subseteq T_{um}$ . Тобто, ця умова означає, що початково маркуватись на цьому шляху може тільки позиція  $S_i$  і ніяка інша. А тому:

$$(\mathbf{A}_0^*)_{ik} \geq \delta, \quad (9.20)$$

оскільки елементи матриці  $\mathbf{A}_0^*$  є шляхами з максимальною вагою між усіма існуючими позиціями, а знак нерівності в (9.20) враховує можливість існування паралельного шляху з більшою вагою.

Для кожної жорсткої умови управління до матриці  $\mathbf{K}_0$  вводиться елемент

$$(\mathbf{K}_0)_{mj} = \delta^{-1} \otimes \tau \otimes \lambda_s^{-K}.$$

Оскільки на передобласть  $S_k$  впливає управління  $u_m$ , для матриці  $\mathbf{B}$  має місце твердження:

$$b_{km} = 0,$$

а це означає, що шлях в  $G(\mathbf{A}_0)$  буде довшим на одне ребро, яке зумовлене відповідною логічною умовою управління.



Узагальнюючи для всіх логічних умов, маємо таке:

$$\left( (\mathbf{A}_0 \oplus \mathbf{BK}_0(\lambda_s))^* \right)_{ij} \geq \delta \otimes 0 \otimes \delta^{-1} \otimes \tau \otimes \lambda_s^k = \tau \otimes \lambda_s^k. \quad (9.21)$$

Таким чином в графі  $G(\mathbf{A}_0 \oplus \mathbf{BK}_0(\lambda_s))$  існує шлях від  $k_j$  до  $k_i$ . Вузол  $k_h$  повинен відповідати початково маркованому положенню  $S_h$ . Окрім того, в графі  $G(\mathbf{A}_0 \oplus \mathbf{BK}_0(\lambda_s))$  існує як мінімум один шлях із  $k_h$  до  $k_j$ , а це означає, що існує також шлях із  $k_h$  до  $k_i$ . Максимальна вага шляху із  $k_h$  до  $k_j$  буде  $u_{jh}$ , а із  $k_h$  до  $k_i - u_{jh}$ , відповідно. Оскільки шлях із  $k_h$  до  $k_i$  може проходити через  $k_j$ , а вага шляху із  $k_j$  до  $k_i$  визначається як  $\tau \otimes \lambda_s^k$ . Для максимальної ваги всіх шляхів із  $k_h$  до  $k_j$  має місце обмеження:

$$\omega_{ih} - \omega_{jh} \geq \tau \otimes \lambda_s^k.$$

Звідси можна стверджувати, що в  $G(\mathbf{M}_s)$  з'являються відповідні ребра, які входять до шляхів, що ведуть із  $k_i$  до  $k_j$  зі зміненою вагою:

$$x_i(k) \geq \tau \otimes \lambda_s^k \otimes x_j(k).$$

Цей вираз повністю підтверджує, що рівняння (9.19) завжди має розв'язок.

Для знаходження власного значення  $\lambda_s$  і власного вектора керованої системи можна використовувати різні ітераційні процедури, в основі яких лежать раніше розглянуті алгоритми (підрозділ 3.3).

Приклад графа, що ілюструє синтез управління з жорсткими умовами, наведено на рис. 9.3. Програми обчислень наведені у додатку, лістинг 7.

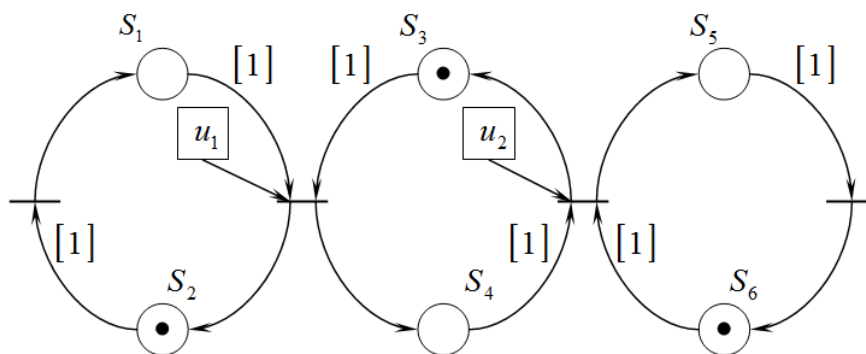


Рисунок 9.3 – Граф синхронізації з жорсткими умовами управління

Для наведеного на рис. 9.3 графа відповідні матриці матимуть вигляд:

$$\mathbf{A}_0 = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 1 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & 1 & \varepsilon & \varepsilon \\ 1 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & 1 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & 1 & \varepsilon \end{bmatrix}, \mathbf{A}_1 = \begin{bmatrix} \varepsilon & 1 & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & 1 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 1 \\ \varepsilon & \varepsilon & 1 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 1 \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{bmatrix}, \mathbf{B} = \begin{bmatrix} \varepsilon & \varepsilon \\ 0 & \varepsilon \\ \varepsilon & 0 \\ 0 & \varepsilon \\ \varepsilon & 0 \\ \varepsilon & \varepsilon \end{bmatrix}.$$

За допомогою \$\mathbf{A}\_0^\*\$ і \$\mathbf{A}\_1\$ розраховується матриця \$\mathbf{M}\$:

$$\mathbf{M} = \begin{bmatrix} \varepsilon & 1 & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & 2 & 1 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & 3 & 2 & \varepsilon & \varepsilon & 1 \\ \varepsilon & 2 & 1 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & 3 & 2 & \varepsilon & \varepsilon & 1 \\ \varepsilon & 4 & 3 & \varepsilon & \varepsilon & 2 \end{bmatrix}.$$

Для матриці \$\mathbf{M}\$ розраховується власне значення \$\lambda = 2\$, циклічність \$\rho = 1\$ і власний вектор:

$$\mathbf{v} = [-1 \ 0 \ 1 \ 0 \ 1 \ 2]^T.$$

Власний вектор промасштабований таким чином, що система для \$\mathbf{x}(0)\$ може стартувати з цим власним вектором.

Початково марковані положення  $S_3$  і  $S_6$  знаходяться в передобласті керованих переходів, а для початково маркованої позиції  $S_2$  маємо компоненту  $v_2 = 0$ . Стартуючи з власним вектором, отримуємо випадок, коли у власному стані одночасно маркуються положення  $S_1$  і  $S_3$ , причому  $S_1$  буде маркуватись на один раз більше за  $S_3$ :

$$x_3(k) = x_1(k+1) = 0 \otimes x_1(k+1),$$

звідки  $\tau_{ist} = 0$  і  $K_{ist} = 1$ . Цей стан можна виключити за допомогою жорстких логічних умов управління, які формулюються наступним чином: позиція  $S_3$  повинна маркуватись через  $\tau$  одиниць часу після позиції  $S_1$ , при цьому довжина циклу не повинна змінюватись, тобто  $K = K_{ist} = 1$ . При розрахунках використовується обчислювальна процедура, яка дозволяє зменшити число розрахунків степенів матриці  $(\mathbf{A}_0 \oplus \mathbf{BK}_0(\lambda_s))^*$  при ітераціях шляхів визначення  $(\mathbf{K}_1)_{21} = 2 \otimes \lambda_s^2$  і  $\mathbf{K}_0 = \mathbf{N}$  відповідно. Шляхом використання ітерації можна знайти власне значення  $\lambda_s = 4$  матриці  $\mathbf{M}_s$  і власний вектор:

$$\mathbf{v}_s = [-3 \ 0 \ 3 \ 0 \ 3 \ 4]^T,$$

який показує, що в замкнутій системі виконуються стартові умови і виконуються логічні умови управління.

### 9.5 Інваріантність в ДБС

Якщо керована ДБС заходить у власному стані, то між матрицями зворотного зв'язку  $\mathbf{K}_0$  і  $\mathbf{K}_1$  існує особливе співвідношення. Як показано в [57], окремі елементи в цих матрицях при стаціонарному власному режимі поведінки ДБС є комутативними (взаємозамінними). Так, матриця  $\mathbf{K}_0$  дозволяє розрахувати вектор управління  $\mathbf{u}(k+1)$  із маркованих часових

точок  $\mathbf{x}(k+1)$  поточного циклу  $(k+1)$ . Всі часові точки стану ДБС у власному стані поведінки  $(k+1)$ -о циклу відрізняються від значень в  $k$ -у циклі на величину власного числа  $\lambda_s$  матриці замкненої системи. Для розрахунку матриці  $\mathbf{K}_1$  використовується вектор  $\mathbf{x}(k)$ . Оскільки у стаціонарному власному стані ДБС між  $\mathbf{x}(k+1)$  і  $\mathbf{x}(k)$  існує однозначний зв'язок, можна перерахувати елементи матриці  $\mathbf{K}_1$  так, ніби матриці  $\mathbf{K}_0$  не існує.

Цей висновок базується на наступних математичних твердженнях. Якщо видалити будь-який елемент із матриці зворотного зв'язку  $\mathbf{K}_0$  і збільшити на  $\lambda_s$  структурно відповідний елемент матриці  $\mathbf{K}_1$ , то власне число матриці  $\mathbf{M}_s$  і будь-який власний вектор цієї матриці залишаться незмінними. Наприклад, елементу  $(\mathbf{K}_0)_{ij} = r$  відповідає в графі  $G(\mathbf{A}_0 \oplus \mathbf{BK}_0)$  ребро від вузла  $k_j$  до вузлів, які знаходяться після керованого переходу з управляючим впливом  $u_1$ . В цьому випадку шляхи в графі  $G(\mathbf{A}_0 \oplus \mathbf{BK}_0)$  можуть бути тільки довшими у порівнянні зі шляхами в графі  $G(\mathbf{A}_0)$ . Кожен шлях в графі  $G(\mathbf{A}_0 \oplus \mathbf{BK}_0)$  закінчується в одній із початково маркованих позицій графа синхронізації. Таким чином, в результаті заміни елементів в  $\mathbf{K}_0$  і  $\mathbf{K}_1$  в графі  $G(\mathbf{M}_s)$  може з'явитися додаткове ребро або збільшитися вага існуючого ребра.

Очевидно, що така заміна не повинна змінювати структуру графа синхронізації ДБО, за умови, що з початково маркованого положення  $S_1$  існує шлях, що не проходить через інші марковані позиції. Наступна  $j$ -а позиція переходу однозначно розраховується на підставі  $(\mathbf{K}_0)_{ij}$ . Якщо  $S_j$  початково маркована, то має місце рівняння  $S_1 = S_j$  і вага цього шляху в мережі Петрі буде дорівнювати  $\omega_1$ . Позиція  $S_k$  в післяобласті керованого переходу з управлінням  $u_i$  маркується в залежності від логічних умов

через  $r$  одиниць часу після позиції  $S_j$ . З позиції  $S_k$  існує наступний шлях, який через логічні умови затримки приводить до початково маркованої позиції  $S_2$ . Цей шлях не містить інших початково маркованих позицій і можна стверджувати, що  $S_2 \neq S_1$  оскільки граф  $G(\mathbf{A}_0 \oplus \mathbf{BK}_0)$  не повинен мати цикли. Вага цього субшляху буде становити  $\omega_2$ .

Якщо вага шляху  $\omega_1 + r + \omega_2$  буде більшою за вагу прямого шляху між позиціями  $S_1$  і  $S_2$ , то в графі  $G(\mathbf{M}_s)$  з'являється додаткове ребро, тобто елемент матриці  $(\mathbf{M}_s)_{21} = \omega_1 + r + \omega_2$ .

Оскільки логічна умова керованого переходу враховується в модифікованій матриці зворотного зв'язку  $(\tilde{\mathbf{K}}_1)_{ij} = \lambda_s + r$ , то це означає, що проміжний вузол  $k_j$  належить до жорстко зв'язаних компонентів графа  $G(\tilde{\mathbf{M}}_s)$  і тому завжди має вихідні ребра. Також очевидно, що перший субшлях із  $S_1$  в  $S_2$  розраховується за допомогою  $(\tilde{\mathbf{M}}_s)_{j1}$ , а логічні умови управління і другий субшлях із  $S_k$  до  $S_2$  – за допомогою  $(\tilde{\mathbf{M}}_s)_{2j}$ . Оскільки при цьому структура графа синхронізації не змінюється, для знаходження ваги шляху справедливою є наступна формула:

$$(\tilde{\mathbf{M}}_s)_{j1} + (\tilde{\mathbf{M}}_s)_{2j} = \omega_1 + r + \lambda_s + \omega_2,$$

з якої витікає, що шлях в графі  $G(\tilde{\mathbf{M}}_s)$  завжди на одне ребро довший, а його вага на  $\lambda_s$  одиниць часу більша, ніж у графі  $G(\mathbf{M}_s)$ .

В загальному випадку можливі два варіанти поведінки ДБС:

1. Ребро, вага якого залежить від логічної умови управління, знаходиться в циклі з вагою  $\omega$  і довжиною  $\rho$  у критичному графі  $G^c(\mathbf{M}_s)$ . Цьому циклу відповідає цикл мережі Петрі і по відношенню до нього в

критичному графі  $G^c(\tilde{\mathbf{M}}_s)$  існує цикл вагою  $\tilde{\omega}$ , довжиною  $\tilde{\rho}$ . Тоді має місце наступне співвідношення:

$$\tilde{\lambda}_s = \frac{\tilde{\omega}}{\tilde{\rho}} \Big|_{G(\tilde{\mathbf{M}}_s)} = \frac{\tilde{\omega} + \lambda_s}{\rho + 1} \Big|_{G(\mathbf{M}_s)} = \frac{\omega + \frac{\omega}{\rho}}{\rho + 1} \Big|_{G(\mathbf{M}_s)} = \frac{\omega}{\rho} \Big|_{G(\mathbf{M}_s)} = \lambda_s,$$

яке показує, що власні числа матриць  $\mathbf{M}_s$  і  $\tilde{\mathbf{M}}_s$  однакові. Це означає, що структура мережі Петрі залишається незмінною.

2. Якщо ребро, вага якого залежить від логічної умови управління, не знаходиться в критичному циклі графа  $G^c(\mathbf{M}_s)$ , то має місце наступна нерівність:

$$\frac{\omega}{\rho} \Big|_{G(\mathbf{M}_s)} < \lambda_s. \quad (9.22)$$

З урахуванням, що завжди  $\rho > 0$ , нерівність (9.22) буде мати вигляд:

$$\frac{\omega}{\rho} \Big|_{G(\mathbf{M}_s)} < \frac{\omega + \lambda_s}{\rho + 1} \Big|_{G(\mathbf{M}_s)} < \lambda_s,$$

звідки:

$$\frac{\tilde{\omega}}{\tilde{\rho}} \Big|_{G(\tilde{\mathbf{M}}_s)} = \frac{\omega + \lambda_s}{\rho + 1} \Big|_{G(\mathbf{M}_s)}.$$

Тоді отримуємо:

$$\frac{\tilde{\omega}}{\tilde{\rho}} \Big|_{G(\tilde{\mathbf{M}}_s)} < \lambda_s.$$

Цикл, який не є критичним в  $G(\mathbf{M}_s)$ , не призводить до критичного циклу і у графі  $G(\tilde{\mathbf{M}}_s)$ . Також існує математичний доказ, наприклад в [57],

згідно з яким власні числа і власні вектори матриць  $\mathbf{M}_s$  і  $\tilde{\mathbf{M}}_s$  повністю ідентичні.

Розглянуті математичні викладки можуть бути узагальнені для ДБС зі значною кількістю додаткових умов управління. Всі вони включаються до  $\mathbf{K}_0$  або  $\mathbf{K}_1$  і в подальшому інваріантність в системі може розглядатися як для окремої одиничної логічної умови. Також зрозуміло, що в загальному випадку довжини шляхів в  $G(\mathbf{M}_s)$  і  $G(\tilde{\mathbf{M}}_s)$  можуть бути різними, циклічність цих графів також може бути різною, а тому перехідний процес може відрізнитись. Таким чином, про інваріантність ДБС можна говорити тільки відносно сталого стану поведінки системи зі зворотним зв'язком.

## 9.6 Компенсація збурень в ДБС

### 9.6.1 Компенсація збурень в ДБС з одним власним вектором

В дискретно-безперервних системах, як і в системах інших класів, можуть виникати збурення, які необхідно врахувати при розрахунках систем керування. Формально збурення в системі можна враховувати в алгебраїчній моделі керованого дискретно-безперервного процесу за допомогою введення в рівняння стану додаткового доданка:

$$\mathbf{x}(k+1) = \mathbf{A}_0 \mathbf{x}(k+1) \oplus \mathbf{A}_1 \mathbf{x}(k) \oplus \mathbf{B} \mathbf{u}(k+1) \oplus \mathbf{E} \mathbf{z}(k+1), \quad (9.23)$$

або:

$$\mathbf{x}(k+1) = \mathbf{A}_0^* \mathbf{A}_1 \mathbf{x}(k) \oplus \mathbf{A}_0^* \mathbf{B} \mathbf{u}(k+1) \oplus \mathbf{A}_0^* \mathbf{E} \mathbf{z}(k+1). \quad (9.24)$$

В цих рівняннях матриця  $\mathbf{E}$  показує, на яких позиціях графа синхронізації виникають збурення, що призводять до затримок заняття позицій в порівнянні з часом заняття в некерованому процесі. Вектор  $\mathbf{z}(k)$  включає часи затримок, пов'язаних з дією збурень. Якщо  $z_i(k) = \varepsilon$ , то це означає, що на  $i$ -й позиції збурення відсутнє.

Найбільш суттєвими є збурення в системах, де алгоритм керування формується за жорсткими логічними умовами. В цьому випадку, за наявності збурення порушується власний стан системи і логічні умови керування не виконуються.

Перехідний процес в ДБС, породжений збуреннями, призводить до втрати власного стану.

Ідея зменшення впливу збурень полягає в як найшвидшому поверненні системи у власний стан, де знову почнуть формуватись керуючі впливи з урахуванням логічних умов.

Для досягнення таких властивостей матриця  $\mathbf{M}_s$  замкнутої керованої системи повинна задовольняти як мінімум двом умовам [59]:

- циклічність  $\rho$  кожного жорстко зв'язаного компонента графа  $G^C(\mathbf{M}_s)$  повинна дорівнювати 1, що забезпечує якнайшвидше переведення ДБС у власний стан;
- матриця  $\mathbf{M}_s$  замкнутої системи повинна мати тільки один власний вектор.

Виконання цих умов в замкнутій системі можна забезпечити шляхом внесення додаткових елементів до матриць зворотного зв'язку  $\mathbf{K}_0$  і  $\mathbf{K}_1$ .

Циклічність жорстко зв'язаних компонентів критичних графів  $G^C(\mathbf{M})$  або  $G^C(\mathbf{M}_s)$  тільки тоді більше 1, коли в них немає замкнутих циклів довжиною 1. Тому, для того щоб досягти циклічності довжиною 1, в критичний граф треба примусово внести цикл, довжиною 1. Це можна зробити за допомогою модифікації матриці  $\mathbf{K}_1$ , якщо існує відповідний керований перехід, що належить будь-якому критичному циклу. При цьому необхідно мати на увазі, якщо циклічність графа  $G^C(\mathbf{M})$  некерованої системи більша ніж 1, то для жорстко зв'язаних компонент



необхідний керований перехід не завжди буде знайдено. В цьому випадку мінімізувати час повернення ДБС у власний стан практично неможливо.

Якщо ж в системі циклічність  $G^C(\mathbf{M})$  дорівнює 1, а її значення зростає для графа  $G^C(\mathbf{M}_s)$ , то в цьому випадку жорсткі логічні умови керованого переходу можна за рахунок особливого вибору компонент матриці зворотного зв'язку  $\mathbf{K}_1$  “пом'якшити” і досягти значення циклічності, що дорівнює 1. Необхідною умовою для досягнення зменшення циклічності керованого переходу  $t_s$  з управлінням  $u_m$  є приналежність цього переходу до критичного циклу  $G(\mathbf{M}_s)$ . Цей цикл належить до жорстко зв'язаних компонент графа  $G^C(\mathbf{M}_s)$ , циклічність якого більше за 1. Так як ребро критичного графа є відповідною частиною циклу, то в мережі Петрі існує шлях, який не проходить через вершини, що відповідають початковим умовам маркування. Тобто із вершини  $S_k$  в післяобласті  $t_s$  і до початково маркованого положення  $S_i$  можна потрапити тільки через вершину, яка належить до жорстко зв'язаних компонент графа  $G^C(\mathbf{M}_s)$ . Вага цього шляху  $\omega_{ik}$  розраховується за наступною формулою:

$$\omega_{ik} = \left( (\mathbf{A}_0 \oplus \mathbf{BK}_0)^* \mathbf{Bw} \right)_i, \quad (9.25)$$

де:

$$\mathbf{w} = [w_1 \dots w_p],$$

$p$  – кількість керованих переходів;

$$w_j = \begin{cases} e, & \text{при } j = m \\ \varepsilon, & \text{в інших випадках.} \end{cases}$$

Тоді за допомогою елемента матриці зворотного зв'язку  $\mathbf{K}_1$ , який відповідає  $m$ -му керованому переходу, розрахованому за формулою:

$$(\mathbf{K}_1)_{mi} = \lambda_s - \omega_{ik}, \quad (9.26)$$

додатково вводиться критичний цикл довжиною 1, який розпочинається і закінчується в вузлі  $k_i$ . Завдяки цьому циклічність цього жорстко зв'язаного компонента в графі  $G^C(\mathbf{M}_s)$  дорівнює 1. Власні вектора і власні числа матриці  $\mathbf{M}_s$  при цьому залишаються незмінними.

### 9.6.2 Приклад, що ілюструє компенсацію збурень в ДБС

Припустимо, структура графа синхронізації має вигляд як наведено на рис. 9.3. Програми обчислень наведені у додатку, лістинг 8. Логічні умови керування таким процесом повністю відповідають типовим, які сформульовано раніше в цьому розділі.

Критичний граф некерованої системи  $G^C(\mathbf{M})$  має жорстку зв'язну компоненту. Циклічність цього графа  $\rho = 1$ . Якщо жорстку логічну умову керування реалізувати за допомогою матриці зворотного зв'язку  $\mathbf{K}_0$ , то структура графа  $G(\mathbf{M}_s)$  має вигляд, представлений на рис. 9.4.

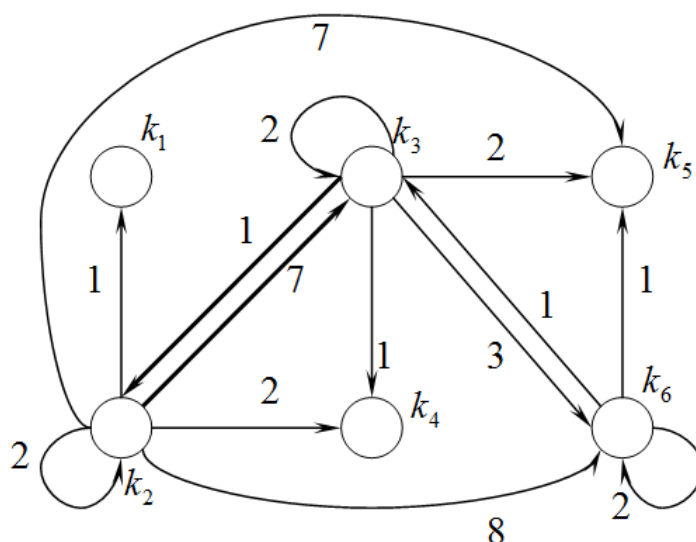


Рисунок 9.4 – Граф  $G(\mathbf{M}_s)$  при трансформуванні жорсткої логічної умови керування за допомогою  $\mathbf{K}_0$

«Жирними» лініями виділено граф  $G^C(\mathbf{M}_s)$ , циклічність якого  $\rho = 2$ .  
Графік перехідного процесу для ДБС в цьому випадку представлено на рис. 9.5.

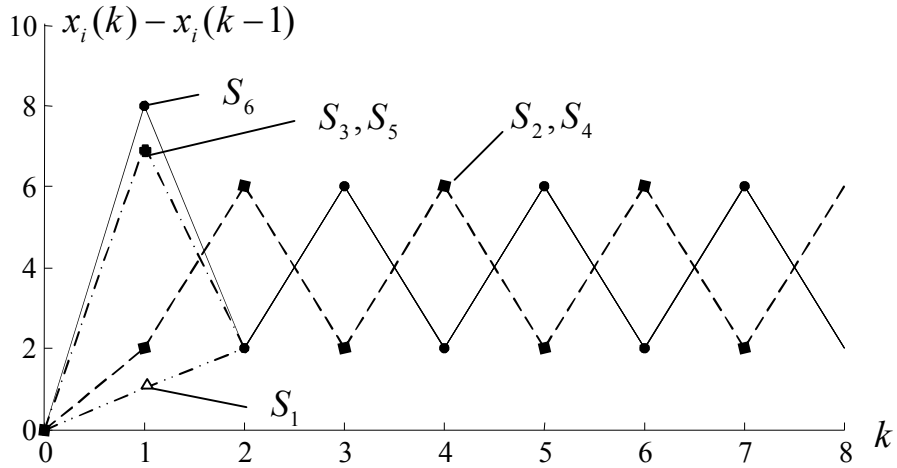


Рисунок 9.5 – Перехідний процес в ДБС з циклічністю  $\rho = 2$

Якщо жорстку логічну умову керування реалізувати за допомогою матриці зворотного зв'язку  $\mathbf{K}_1$ , то граф, що відповідає матриці замкнутої системи  $G(\mathbf{M}_s)$ , матиме вигляд, як представлено на рис. 9.6.

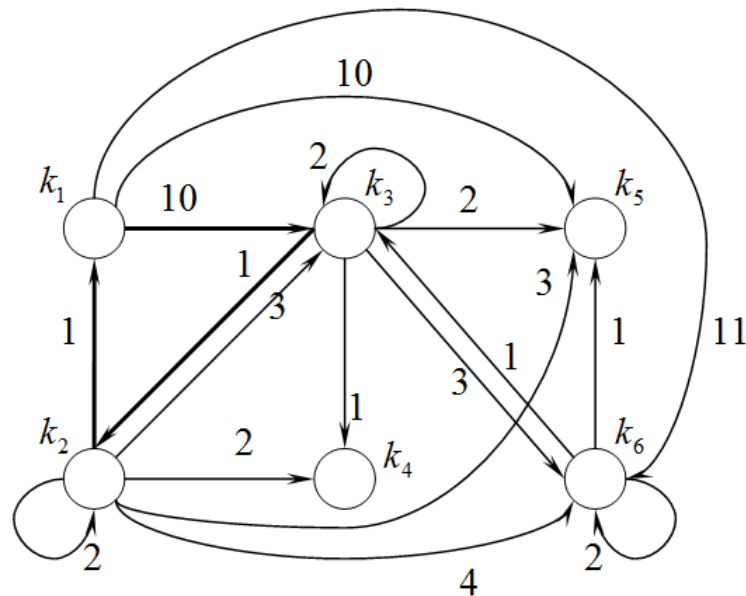


Рисунок 9.6 – Граф  $G(\mathbf{M}_s)$  при трансформуванні жорстких логічних умов керування за допомогою  $\mathbf{K}_1$

Критичний граф  $G^C(\mathbf{M}_s)$  на цьому рисунку також виділений «жирними» лініями. Звідси очевидно, що циклічність цього графа  $\rho = 3$ .

Перехідний процес в системі представлено на рис. 9.7.

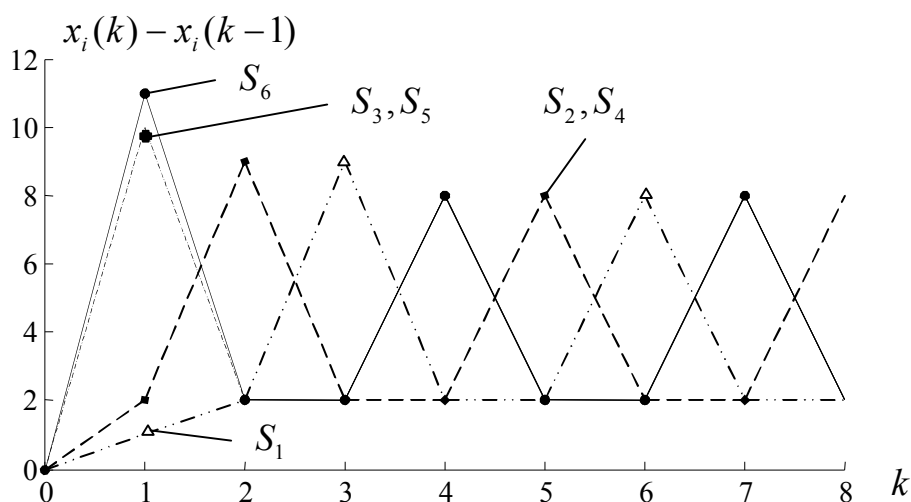


Рисунок 9.7 – Перехідний процес в ДБС з циклічністю  $\rho = 3$

Для компенсації збурень можна застосувати теорію, основи якої наведено в цьому розділі. Для цього задається вектор  $\mathbf{w}$ :

$$\mathbf{w} = [\varepsilon \quad e]^T.$$

В першому випадку

$$(\mathbf{A}_0 \otimes \mathbf{BK}_0)^* \mathbf{Bw} = [\varepsilon \quad \varepsilon \quad e \quad \varepsilon \quad e \quad 1]^T,$$

а в другому випадку:

$$(\mathbf{A}_0^* \mathbf{Bw}) = [\varepsilon \quad \varepsilon \quad e \quad \varepsilon \quad e \quad 1]^T.$$

Таким чином, в обох випадках  $w_{33} = e$  і елемент матриці зворотного зв'язку для керованого переходу  $t_3$  можна визначити як:

$$(\mathbf{K}_1)_{23} = \lambda_s - w_{33} = 4 - 0 = 4.$$

Якщо логічна умова керування реалізується за допомогою  $\mathbf{K}_1$ , маємо:

$$\mathbf{K}_1 = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 10 & \varepsilon & 4 & \varepsilon & \varepsilon & \varepsilon \end{bmatrix},$$

$$(\mathbf{K}_0)_{ij} = e,$$

$$\mathbf{M}_s = \begin{bmatrix} \varepsilon & 1 & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & 2 & 1 & \varepsilon & \varepsilon & \varepsilon \\ 10 & 3 & 4 & \varepsilon & \varepsilon & 1 \\ \varepsilon & 2 & 1 & \varepsilon & \varepsilon & \varepsilon \\ 10 & 3 & 4 & \varepsilon & \varepsilon & 1 \\ 11 & 4 & 5 & \varepsilon & \varepsilon & 2 \end{bmatrix}.$$

Тоді граф, що відповідає замкнутій системі, при трансформації умов в  $\mathbf{K}_1$  і  $\rho = 1$  матиме вигляд, що представлений на рис. 9.8. Циклічність досягнуто за рахунок додаткового контуру у вузлі  $k_3$  вагою 4 і довжиною 1.

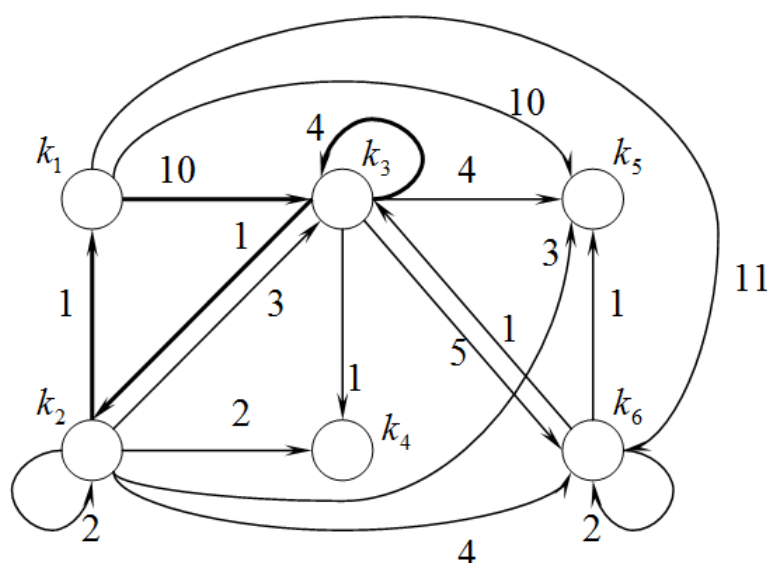


Рисунок 9.8 – Граф  $G(\mathbf{M}_s)$  при трансформуванні жорсткої логічної умови керування в  $\mathbf{K}_1$  і  $\rho = 1$

На рис. 9.9 приведено графік перехідного процесу в ДБС, який ілюструє встановлення власного стану в системі.

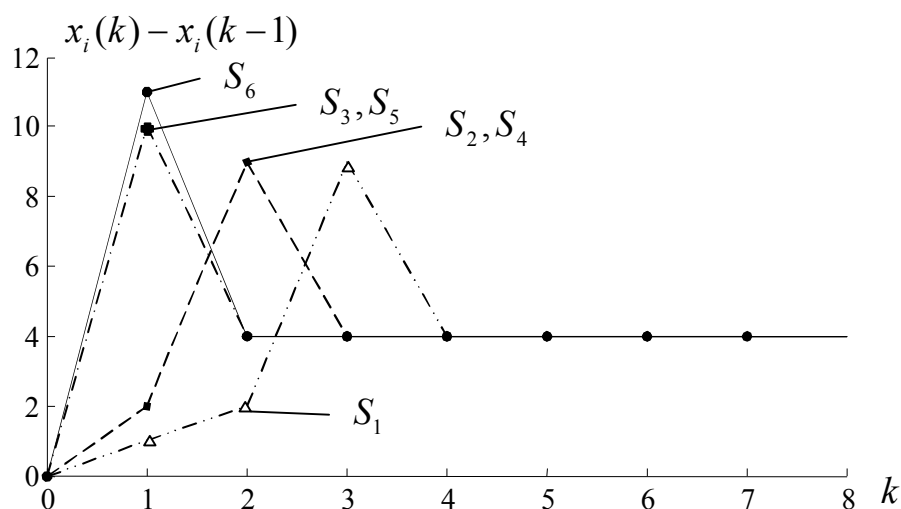


Рисунок 9.9 – Перехідний процес в ДБС з  $\rho = 1$

Якщо логічні умови керування з урахуванням інваріантності трансформувати в  $\mathbf{K}_0$  одержимо наступні перетворення. Оскільки  $\mathbf{K}_1$  має тільки один елемент, відмінний від  $\varepsilon$ , а саме  $(\mathbf{K}_1)_{23} = 4$ , матриця замкненої системи матиме такий вигляд:

$$\mathbf{M}_s = \begin{bmatrix} \varepsilon & 1 & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & 2 & 1 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & 7 & 4 & \varepsilon & \varepsilon & 1 \\ \varepsilon & 2 & 1 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & 7 & 4 & \varepsilon & \varepsilon & 1 \\ \varepsilon & 8 & 5 & \varepsilon & \varepsilon & 2 \end{bmatrix}.$$

Цій матриці відповідає граф, який представлено на рис. 9.10.

Циклічність  $\rho = 1$  досягається знову за рахунок додаткового шляху з вагою 4 і довжиною 1 у вершині  $k_3$ .

Власний вектор ДБС в обох випадках залишається незмінним:

$$\mathbf{v} = [-3 \quad e \quad 3 \quad e \quad 3 \quad 4]^T,$$

і власне число  $\lambda_s = 4$  також відповідає значенню, що було в системі до урахування компенсації збурень.

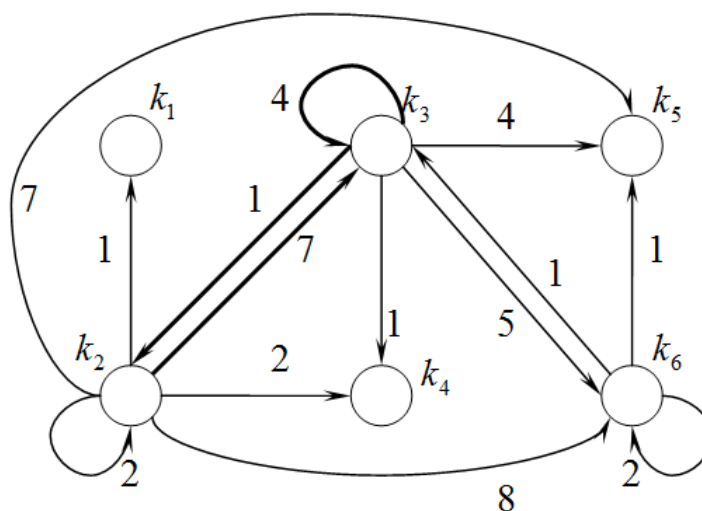


Рисунок 9.10 – Граф  $G(\mathbf{M}_s)$  при трансформації жорстких логічних умов керування в  $\mathbf{K}_0$  і  $\rho = 1$

### 9.6.3 Компенсація збурень в ДБС з декількома власними векторами

Якщо в керованій системі існує декілька власних векторів, то критичний граф  $G^c(\mathbf{M}_s)$  має відповідне число жорстко зв'язаних компонент. Для того щоб виключити хоча б один вектор, необхідно шляхом додаткового внесення компонент в матрицю зворотного зв'язку прив'язати як мінімум дві жорстко зв'язаних компоненти до одного спільного елемента матриці зворотного зв'язку.

Будь-які внесення елементів, відмінних від  $\varepsilon$ , в матрицю зворотного зв'язку підвищують вагу існуючих ребер або призводять до появи нових ребер в графі  $G(\mathbf{M}_s)$ . Через ці ребра з іншими ребрами графа необхідно побудувати нові критичні цикли. Новий цикл повинен включати вузли як мінімум із двох різних компонент, що об'єднуються. Таким способом

повинні бути охоплені всі нові цикли і всі компоненти для того, щоб побудувати єдину жорстко зв'язану компоненту. За допомогою відповідного вибору додаткових внесень у будь-яку із матриць зворотного зв'язку можливо досягти того, що один з власних векторів, які об'єднуються, залишається незмінним, а інші або змінюються, або зникають. Іншими словами, шляхом внесення в одну з матриць зворотного зв'язку додаткових елементів множина із  $n$  власних векторів  $v_{si}$ ,  $i \in \{1, 2, 3, \dots, n\}$  матриці  $\mathbf{M}_s$  умовно розділяється на три підмножини:

- підмножина власних векторів, які залишаються незмінними –  $\Pi_1$ .

Якщо ця підмножина має один елемент, то система після збурення повертається у власний стан, який був до моменту виникнення збурення;

- підмножина власних векторів, які зникають –  $\Pi_2$ . Їх кількість дорівнює  $|\Pi_2|$ ;

- підмножина власних векторів, які трансформуються –  $\Pi_3$ .

Таким чином,

$$\Pi_1 \cup \Pi_2 \cup \Pi_3 = \{1, 2, 3, \dots, n\}$$

Власні вектори системи не можуть бути апріорі поділені на три підмножини, скоріше необхідно знайти необхідні керовані переміщення і відповідні позиції в графі синхронізації, які забезпечують бажаний розподіл.

Умови розподілу векторів при додатковому внесенні значущих елементів в будь-яку із матриць зворотного зв'язку базуються на наступних теоретичних постулатах і математичних судженнях.

Для кожного із  $n$  власних векторів  $v_{si}$  матриці  $\mathbf{M}_s$  можна вибрати деякий вузол  $k_y$ , який жорстко зв'язаний з графом  $G^c(\mathbf{M}_s)$ :

$$y = f(i).$$



Кожний власний вектор можна промасштабувати таким чином:

$$(v_{si})_{f(i)} = 0.$$

Розподіл  $n$  власних векторів на три підмножини виконується наступним чином:

- вершина  $S_k$  графа синхронізації  $G(\mathbf{M}_s)$  розміщена в післяобласті керованого переходу  $t_s$  з управлінням  $u_m$ . Ця вершина також належить критичному циклу мережі Петрі, що включає жорстко зв'язні компоненти графа  $G^c(\mathbf{M}_s)$ . Цим жорстко зв'язним компонентам відповідає власний вектор  $v_{s\alpha}$ , який належить до підмножини  $\Pi_1$  ( $\alpha \in \Pi_1$ ).

- для будь-якого фіксованого  $j$  мають місце наступні твердження:

$$\forall \beta \in \Pi_1 : \omega = (v_{s\beta})_k - (v_{s\beta})_j \geq -\lambda_s, \quad (9.27)$$

$$\forall \beta \in \Pi_1, \forall \mu \in \Pi_2 \cup \Pi_3 : \omega = (v_{s\beta})_k - (v_{s\beta})_j > (v_{s\mu})_k - (v_{s\mu})_j, \quad (9.28)$$

$$\forall \mu \in \Pi_2 : \omega = (v_{s\alpha})_{f(\mu)} - (v_{s\alpha})_j + (v_{s\mu})_j = 0. \quad (9.29)$$

З урахуванням того, що граф  $G(\mathbf{A}_0 \oplus \mathbf{BK}_0)$  залишається вільним від циклів, зворотний зв'язок трансформується наступним чином:

$$(\mathbf{K}_0)_{mj} = \omega \text{ для } \omega \geq 0,$$

або

$$(\mathbf{K}_1)_{mj} = \lambda_s \otimes \omega \text{ для } 0 > \omega \geq -\lambda_s.$$

Така керована система має власні вектори  $v_{s\beta}$  з  $\beta \in \Pi_1$ . При цьому  $\Pi_3$  – підмножина трансформованих власних векторів.

При внесенні змін до матриці зворотного зв'язку необхідно пам'ятати, що зміни не повинні призводити до появи від'ємних елементів в цих матрицях.

Згідно з раніше доведеним твердженням, що при  $(\mathbf{K}_0)_{mj} = \omega$  та  $(\mathbf{K}_1)_{mj} = \lambda_s \otimes \omega$ , власні вектори системи залишаються незмінними. Тому, достатньою умовою невід'ємності елементів матриці зворотного зв'язку є невід'ємність  $\omega \geq 0$ .

Так як всі власні вектори матриці замкнутої системи, що пов'язані з підмножиною  $\Pi_1$ , означають власний стан системи, будь-який стан  $S_k$  маркується через  $\lambda_s$  одиниць часу після стану  $S_j$ . Цей факт очевидної поведінки замкнутої системи відображається на графі синхронізації шляхом додаткового внесення в  $(\mathbf{K}_0)_{mj} = \omega$ . Звідси витікає твердження, що власні вектори підмножини  $\Pi_1$  залишаються незмінними і власне число матриці замкнутої системи не збільшується.

В [34] доведено, що кожна жорстко зв'язана компонента графа  $G^c(\mathbf{M}_s)$  пов'язана з одним відповідним власним вектором. Необхідно, щоб зникли власні вектори підмножини  $\Pi_2$ . Тобто при внесенні елементів в матрицю  $\mathbf{K}_0$  із жорстко зв'язаних компонентів цих векторів підмножини  $\Pi_2$  і жорстко зв'язаних компонентів векторів підмножини  $\Pi_1$  треба сформулювати спільну жорстко зв'язану компоненту. З математичної точки зору достатньо показати, що для  $\alpha \in \Pi_1$  і будь-якого  $\mu \in \Pi_2$  існує критичний цикл, який включає як позицію  $S_{f(\alpha)}$ , якій належить жорстко зв'язана компонента  $v_{s\alpha}$ , так і позицію  $S_{f(\mu)}$ , якій належить жорстко зв'язана компонента  $v_{s\mu}$ .

Для цього розглянемо нормовану матрицю  $\mathbf{M}_{sn} = \lambda_s^{-1} \mathbf{M}_s$ . Вагу шляху між двома вузлами  $G(\mathbf{M}_{sn})$  можна визначити за допомогою власних векторів, які також являються нормованими.

Розглянемо три шляхи в мережі Петрі, які в  $G(\mathbf{M}_{sn})$  складають цикл, що включає позиції  $S_{f(\alpha)}$  і  $S_{f(\mu)}$ . Вага цього циклу визначається за допомогою ваги окремих шляхів:

- шлях із  $S_{f(\alpha)}$  в  $S_{f(\mu)}$  має в  $G(\mathbf{M}_{sn})$  максимальну вагу  $(v_{s\alpha})_{f(\mu)}$ ;
- шлях із позиції  $S_{f(\mu)}$  в положення  $S_k$  має в  $G(\mathbf{M}_{sn})$  максимальну вагу  $(v_{s\mu})_k$ . Якщо мають місце додаткові внесення в матрицю зворотного зв'язку  $\mathbf{K}_0$ , його вага підвищується на величину  $(v_{s\mu})_j + (v_{s\alpha})_k + (v_{s\alpha})_j$ , оскільки при внесенні має місце:

$$(v_{s\mu})_j + \omega = (v_{s\mu})_j + (v_{s\alpha})_k - (v_{s\alpha})_j > (v_{s\mu})_k;$$

- для визначення ваги шляху із  $S_k$  в  $S_{f(\alpha)}$  можна скористатись тим фактом, що  $S_k$  належить до критичного циклу мережі Петрі. Вага цього циклу в графі, що відповідає нормованій матриці, дорівнює  $e$ . Якщо цикл розкладається на дві складові, то вага відповідно розподіляється порівну. Для шляху, що тут розглядається, вага дорівнює величині  $(v_{s\alpha})_k$ . Загальна вага цього циклу згідно з (9.27) – (9.29) дорівнює:

$$(v_{s\alpha})_{f(\mu)} + (v_{s\mu})_j + (v_{s\alpha})_k - (v_{s\alpha})_j - (v_{s\alpha})_k = 0,$$

тобто цикл є критичним. А звідси витікає, що при додатковому внесенні елементів в матрицю зворотного зв'язку в системі з'являються жорстко зв'язані компоненти в графі  $G^c(\mathbf{M}_s)$ , які включають жорстко зв'язані компоненти власних векторів підмножини  $\Pi_2$ .

Умови (9.27) – (9.29) використовуються тільки для визначення підмножини  $\Pi_1$ ,  $\Pi_2$  і  $\Pi_3$ , тобто за допомогою цих співвідношень можна сказати, які власні вектора системи зберігаються, а які знищуються або трансформуються. Для того, щоб цілеспрямовано окремі вектора зберегти,

а деякі знищити, необхідно відповідним чином вибрати позиції  $S_j$  і  $S_k$  в графі синхронізації, відносно яких вносяться зміни в матрицю зворотного зв'язку. Загальнотеоретичний підхід до розв'язання цієї задачі досить складаний, тому в рамках цієї монографії наведено приклад 9.2, який ілюструє, як за рахунок зміни відповідних позицій розгляду можна впливати на властивості керованої ДБС.

Приклад 9.2.

Припустимо керованій ДБС відповідає граф синхронізації з декількома критичними циклами (рис. 9.11).

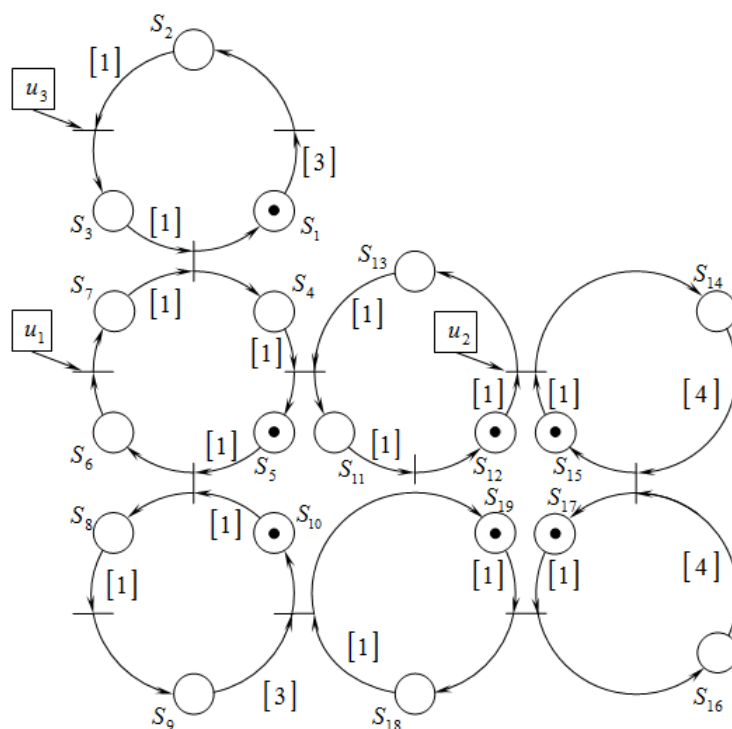


Рисунок 9.11 – Граф синхронізації ДБС з трьома власними векторами в матриці замкнутої системи

Власне число мережі Петрі, що представлена на рис. 9.11,  $\lambda = 5$ . Відповідний критичний граф  $G^c(\mathbf{M})$  включає три жорстко зв'язані компоненти.

Перша компонента  $k_1$  пов'язана з початково маркованою позицією  $S_1$ , друга компонента  $k_{10}$ , що відповідає позиції  $S_{10}$  графа синхронізації, і третя компонента включає дві вершини  $k_{15}$  і  $k_{17}$ , які відповідно пов'язані з позиціями  $S_{15}$  і  $S_{17}$ . Таким чином, матриця  $\mathbf{M}$  має три власні вектори. Для масштабування цих векторів встановлено:

$$f(1)=1; f(2)=10; f(3)=15.$$

При цьому отримано масштабовані власні вектори ДБС:

$$\mathbf{v}_1 = [0 \ -2 \ -1 \ 0 \ 1 \ -3 \ -2 \ -3 \ -2 \ 1 \ 1 \ 2 \ -2 \ -2 \ 2 \ -2 \ 2 \ -2 \ 1]^T,$$

$$\mathbf{v}_2 = [-2 \ -4 \ -3 \ -2 \ -1 \ -4 \ -3 \ -4 \ -3 \ 0 \ -1 \ 0 \ -4 \ -4 \ 0 \ -4 \ 0 \ -4 \ 0]^T,$$

$$\mathbf{v}_3 = [-5 \ -7 \ -6 \ -5 \ -3 \ -7 \ -6 \ -7 \ -6 \ -3 \ -3 \ -2 \ -4 \ -4 \ 0 \ -4 \ 0 \ -4 \ -3]^T.$$

Щоб продемонструвати, як за допомогою вибору  $S_j$  і  $S_k$  можна міняти кількість власних векторів і тим самим змінювати властивості ДБС, розглянемо декілька варіантів:

$$1. S_j = S_2 \text{ і } S_k = S_{14}.$$

В цьому випадку  $S_{14}$  належить до критичного циклу мережі Петрі, який пов'язаний з власним вектором  $\mathbf{v}_3$ . Тому цей вектор не буде змінюватись і  $\alpha = 3$ . На керований перехід передобласті  $S_{14}$  впливає управління  $u_2$ , тому  $m = 2$ . Перевірка умов (9.27) і (9.28) дозволяє розділити власні вектори між підмножинами  $\Pi_1$ ,  $\Pi_2$  і  $\Pi_3$  наступним чином:

$$(\mathbf{v}_1)_{14} - (\mathbf{v}_1)_2 = (-2) - (-2) = 0,$$

$$(\mathbf{v}_2)_{14} - (\mathbf{v}_2)_2 = (-4) - (-4) = 0,$$

$$(\mathbf{v}_3)_{14} - (\mathbf{v}_3)_2 = (-4) - (-7) = 3.$$

В результаті маємо  $\omega = 3$ ,  $\Pi_1 = \{3\}$  і  $\Pi_2 \cup \Pi_3 = \{1,2\}$ . Для визначення підмножин  $\Pi_2$  і  $\Pi_3$  використовується умова (9.29):

$$\begin{aligned} (\mathbf{v}_3)_1 - (\mathbf{v}_3)_2 + (\mathbf{v}_1)_2 &= 0; \\ (\mathbf{v}_3)_{10} - (\mathbf{v}_3)_2 + (\mathbf{v}_2)_2 &= 0. \end{aligned}$$

Звідси витікає  $\Pi_2 = \{1,2\}$  і  $\Pi_3 = 0$ . Тому, якщо в матриці  $\mathbf{K}_0$  виконано внесення

$$(\mathbf{K}_0)_{22} = \omega = 3,$$

то керована таким способом система матиме тільки один власний вектор  $\mathbf{v}_3$ .

$$2. S_j = S_{16} \text{ і } S_k = S_3.$$

В цьому випадку на керований перехід, що має позицію  $S_3$  в після області, впливає управління  $u_3$  ( $m=3$ ). Критичний цикл, до якого належить  $S_3$ , жорстко зв'язаний з власним вектором  $\mathbf{v}_1$ . Перевірка умов (9.27) і (9.28):

$$\begin{aligned} (\mathbf{v}_1)_3 - (\mathbf{v}_1)_{16} &= 1, \\ (\mathbf{v}_2)_3 - (\mathbf{v}_2)_{16} &= 1, \\ (\mathbf{v}_3)_3 - (\mathbf{v}_3)_{16} &= -2, \end{aligned}$$

дозволяє отримати  $\omega = 1$ ,  $\Pi_1 = \{1,2\}$  і  $\Pi_2 \cup \Pi_3 = \{3\}$ . Лише вектор  $\mathbf{v}_1$  жорстко зв'язаний з компонентами критичного циклу, тому  $\alpha = 1$ . За допомогою умови (9.29) можна установити, до якої підмножини належить  $\mathbf{v}_3$ :

$$(\mathbf{v}_1)_{15} - (\mathbf{v}_1)_{16} + (\mathbf{v}_3)_{16} = 0,$$

звідки  $\Pi_2 = \{3\}$ , а  $\Pi_3 = 0$ . Керована система за допомогою визначення

$$(\mathbf{K}_0)_{3,16} = \omega = 1,$$

матиме два власні вектора  $\mathbf{v}_1$  і  $\mathbf{v}_2$ , а  $\mathbf{v}_3$  в цьому випадку буде знищено.

$$3. S_j = S_6 \text{ і } S_k = S_3.$$

В цьому випадку позиція  $S_3$  залишається такою ж, як і в попередньому випадку, а тому  $\alpha = 1 \in P_1$ . Застосувавши умови (9.27) – (9.29), отримаємо розподіл:  $P_1 = \{1\}$ ,  $P_2 = \{2\}$  і  $P_3 = \{3\}$ , при цьому  $\omega = 2$ . Якщо в керованій системі зробити визначення  $(\mathbf{K}_0)_{3,6} = \omega = 2$ , то в системі залишається власний вектор  $\mathbf{v}_1$  і новий власний вектор  $\mathbf{v}_{new}$ :

$$\mathbf{v}_{new} = [3 \ 1 \ 2 \ 3 \ 4 \ 0 \ 1 \ 0 \ 1 \ 4 \ 4 \ 5 \ 3 \ 3 \ 7 \ 3 \ 7 \ 3 \ 4]^T.$$

## 10 Приклад синтезу системи керування хімічним реактором змішування компонентів

Хіміко-технологічні об'єкти змішування компонентів, незважаючи на досить просту структуру, дозволяють проілюструвати характерні особливості синтезу системи керування ДБО на базі загальних підходів, характерних для сучасної теорії управління. Технологічний процес, що розглядається в цьому розділі та його детальна характеристика наведені в [48]. Спрощена схема процесу має вигляд, представлений на рис. 10.1.

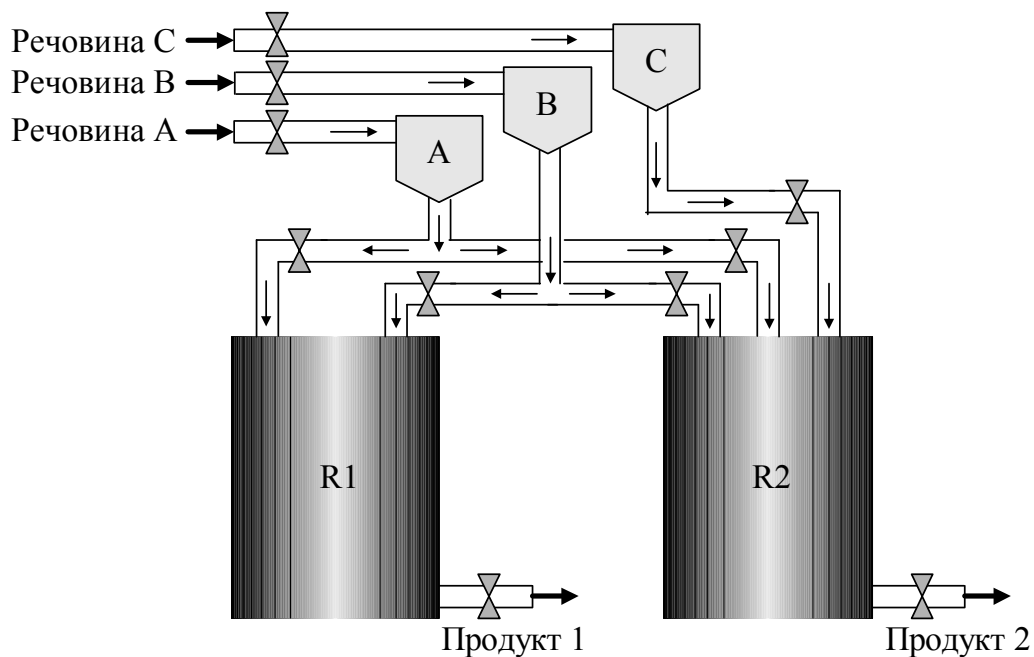


Рисунок 10.1 – Схема хімічного реактора змішування компонент

Для вироблення вихідного продукту 1 в реактор  $R_1$  спочатку повинен бути дозований деякий об'єм речовини  $A$ , потім дозований об'єм речовини  $B$ . Після дозування обох компонентів реактор запускається – відбувається змішування і через деякий час вихідний продукт може



використовуватись для подальшої переробки. Для вихідного продукту 2 в реактор  $R_2$  речовини повинні подаватись в послідовності – речовина  $C$ , потім  $B$  і лише потім  $A$ . Після змішування протягом деякого часу на виході отримують готовий продукт 2. Особливість технологічного процесу полягає в тому, що речовини  $A$  і  $B$  використовуються в обох продуктах і для нормальної роботи дозаторів не можуть одночасно забезпечувати дозування для обох продуктів. Це обмеження повинно враховуватись при розробці алгоритму керування процесом.

Модель, яка описує поведінку хімічного реактора може бути представлена у вигляді графа синхронізації мережі Петрі, наведеного на рис. 10.2. Відповідна інтерпретація елементів мережі наведена в таблиці 10.1.

**Таблиця 10.1 – Інтерпретація елементів мережі**

Позиція	Інтерпретація позиції	Часові моменти	Інтерпретація часового моменту
$S_1$	$R_1$ – порожній	$t_1$	початок дозування речовини $A$ в $R_1$
$S_2$	Дозування речовини $A$ в $R_1$	$t_2$	кінець дозування речовини $A$ в $R_1$
$S_3$	Очікування	$t_3$	початок дозування речовини $B$ в $R_1$
$S_4$	Дозування речовини $B$ в $R_1$	$t_4$	кінець дозування речовини $B$ в $R_1$
$S_5$	Змішування, реакція в $R_1$ , опорожнення	$t_5$	кінець опорожнення $R_1$
$S_6$	$R_2$ – порожній	$t_6$	початок дозування речовини $C$ в $R_2$
$S_7$	Дозування речовини $C$ в $R_2$	$t_7$	кінець дозування речовини $C$ в $R_2$
$S_8$	Очікування	$t_8$	початок дозування речовини $B$ в $R_2$

Продовження таблиці 10.1

Позиція	Інтерпретація позиції	Часові моменти	Інтерпретація часового моменту
$S_9$	Дозування речовини $B$ в $R_2$	$t_9$	кінець дозування речовини $B$ в $R_2$
$S_{10}$	Очікування	$t_{10}$	початок дозування речовини $A$ в $R_2$
$S_{11}$	Дозування речовини $A$ в $R_2$	$t_{11}$	кінець дозування речовини $A$ в $R_2$
$S_{12}$	Змішування, реакція, опорожнення $R_2$	$t_{12}$	кінець опорожнення $R_2$

За вихідними ребрами позицій  $S_1, S_3, S_6, S_8, S_{10}$  встановлена часова оцінка, що дорівнює 0, оскільки в цих позиціях процес може знаходитись скільки завгодно довго (вони не є часозалежними).

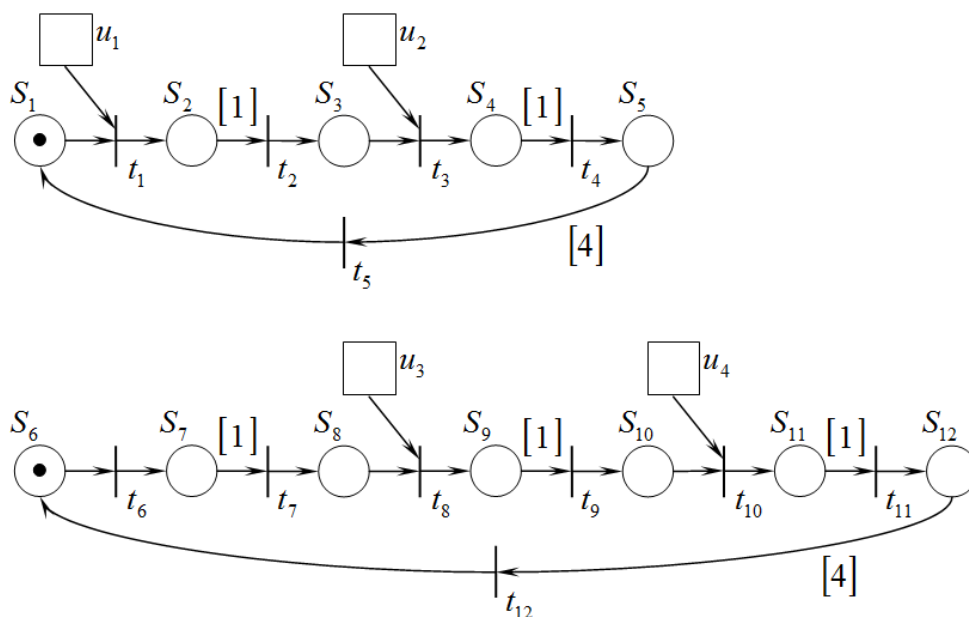


Рисунок 10.2 – Модель хімічного реактора у термінах мережі Петрі

Для того, щоб гарантувати індивідуальне використання дозаторів  $A$  і  $B$  для кожного вихідного продукту необхідно, щоб позиції  $S_2$  і  $S_{11}$ , а також  $S_4$  і  $S_9$  маркувались не одночасно. Для таких і подібних об'єктів

відомі різноманітні емпіричні стратегії управління, які забезпечують виконання наведених вище обмежень [47]. Здебільшого запропоновані алгоритми управління використовують логічно-евристичні підходи і не можуть розглядатись як універсальна теорія синтезу управління. Використання апарату Мах-Plus алгебри дозволяє формалізувати процес синтезу з підходами які характерні для класичної теорії управління.

Методика синтезу управління полягає в наступному. Модель некерованого об'єкта розглядається як моделі двох незалежних процесів виробництва продукту 1 і продукту 2. Фактично ми маємо справу з двома підмережами Петрі. Для кожної з цих двох підмереж може бути знайдена матриця  $\mathbf{M}$ . Програми обчислень наведені у додатку, лістинг 9.

Для першої і другої підмережі матриці динаміки  $\mathbf{A}^{(1)}$  і  $\mathbf{A}^{(2)}$  відповідно мають вигляд:

$$\mathbf{A}^{(1)} = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & 4 \\ e & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & 1 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & e & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & 1 & \varepsilon \end{bmatrix} \text{ і } \mathbf{A}^{(2)} = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 4 \\ e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & 1 & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & 1 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 1 \end{bmatrix}.$$

З урахуванням того, що позиції  $S_1$  та  $S_6$  є початково маркованими, за допомогою трансформуючих матриць можуть бути розраховані матриці  $\mathbf{A}_0^{(i)}$  і  $\mathbf{A}_1^{(i)}$  ( $i = 1, 2$ ):

$$\mathbf{A}_0^{(1)} = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & 4 \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & 1 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & e & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & 1 & \varepsilon \end{bmatrix} \text{ і } \mathbf{A}_1^{(1)} = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ e & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{bmatrix},$$

$$\mathbf{A}_0^{(2)} = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 4 \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & 1 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & 1 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 1 & \varepsilon \end{bmatrix} \text{ і } \mathbf{A}_1^{(2)} = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{bmatrix}.$$

За формулою (8.16) обчислюються матриці  $\mathbf{M}_1$  і  $\mathbf{M}_2$  для кожної з підмереж:

$$\mathbf{M}_1 = \begin{bmatrix} 6 & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ e & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 1 & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 1 & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 2 & \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{bmatrix} \text{ і } \mathbf{M}_2 = \begin{bmatrix} 7 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 1 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 1 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 2 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 2 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 3 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{bmatrix}.$$

Матриця  $\mathbf{M}_1$ , знайдена для виробництва продукту 1, має власне число  $\lambda_1 = 6$  і відповідний власний вектор для позицій з  $S_1$  до  $S_5$  має вид:

$$\mathbf{v}_1 = [6 \ 0 \ 1 \ 1 \ 2]^T.$$

Матриця  $\mathbf{M}_2$  яка описує процес виробництва продукту 2, має власне число  $\lambda_2 = 7$  і власний вектор для позицій від  $S_6$  до  $S_{12}$ :

$$\mathbf{v}_2 = [7 \ 0 \ 1 \ 1 \ 2 \ 2 \ 3]^T.$$

Розглянута в попередніх розділах методика синтезу припускає, що мережа Петрі завжди є жорстко зв'язаною. В нашому випадку ми маємо дві незв'язні підмережі з різними власними числами матриць динаміки. Але, не дивлячись на таку особливість об'єкта, задача синтезу може бути вирішена.

При реалізації методики необхідно буде виконати деякі додаткові умови.

Особливим способом будується загальна матриця мережі Петрі:

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_1 & \mathbf{N} \\ \mathbf{N} & \mathbf{M}_2 \end{bmatrix}.$$

Знаходяться матриці  $\mathbf{A}_0$ ,  $\mathbf{A}_1$  і  $\mathbf{B}$  для всієї системи. Граф  $G(\mathbf{M})$  має дві жорстко зв'язані компоненти. На першому етапі синтезу алгоритму керування ці компоненти об'єднуються за допомогою двох елементів матриці зворотного зв'язку  $\mathbf{K}_0$ . Вибір цих двох елементів матриці  $\mathbf{K}_0$  повинен не сповільнювати процес і забезпечувати коректне використання дозаторів  $A$  і  $B$  для кожного з вихідних продуктів.

Необхідно, щоб елементи зворотного зв'язку забезпечували зв'язок між жорстко зв'язними компонентами  $G(\mathbf{M})$ , так як цим самим і досягається, що в  $G(\mathbf{M}_s)$  присутня тільки одна жорстко зв'язна компонента.

Оскільки при виробництві продукту 2 матриця динаміки цієї підмережі має більше власне число, раціонально дозатор для речовини  $B$  спочатку задіяти в цьому виробництві, щоб не збільшувати власне число. В результаті може бути сформульована проста логічна умова управління: позиція  $S_{10}$  повинна бути зайнятою раніше, ніж відбудеться маркування позиції  $S_4$ . Так як позиція  $S_{10}$  вже вказує на момент закінчення дозування речовини  $B$  в  $R_2$ , бажана часова дистанція між зайняттями цих позицій  $\tau = 0$ . Позиція  $S_4$  маркується безпосередньо після включення керованого переходу  $t_3$ , на який впливає управління  $u_2$  і при цьому  $\delta = 0$ . Тому відповідний елемент матриці зворотного зв'язку:

$$(\mathbf{K}_0)_{2,10} = 0.$$

Цьому елементу відповідає ребро яке з'єднує жорстко зв'язані компоненти графа  $G(\mathbf{M})$  і які пов'язані з циклом виробництва продукту 2. Для того, щоб реалізувати ребро, яке зв'язувало б компоненти в зворотному напрямку необхідно сповільнити зайняття позиції  $S_{11}$  по відношенню до  $S_3$ . При цьому дозатор речовини  $A$  попередньо повинен використовуватись для продукту 1. Оскільки  $S_3$  знову вказує на момент закінчення процесу дозування речовини  $A$  в  $R_1$ , то часовий інтервал між часовими моментами маркування цих позицій досягається при  $\tau = 0$ . Так, за допомогою управління  $u_4$  можна впливати на часову точку заняття позиції  $S_{11}$  ( $\delta = 0$ ). При цьому:

$$(\mathbf{K}_0)_{4,3} = 0.$$

За рахунок цих двох внесень в матрицю зворотного зв'язку досягається бажана поведінка керованого об'єкта. Ця поведінка описується власним вектором  $\mathbf{v}_s$  матриці  $\mathbf{M}_s = (\mathbf{A}_0 \oplus \mathbf{BK}_0)^* \mathbf{A}_1$ . В нашому випадку для цієї матриці власне число  $\lambda_s = 7$ , а власний вектор:

$$\mathbf{v}_s = [7 \ 0 \ 1 \ 2 \ 3 \ 7 \ 0 \ 1 \ 1 \ 2 \ 2 \ 3]^T.$$

Аналіз числових компонент цього вектора показує, що дозатор речовини  $A$  використовується спочатку для реактора  $R_1$  ( $v_{s2} = 0$ ), а потім для реактора  $R_2$  ( $v_{s11} = 2$ ). Дозатор речовини  $B$  використовується спочатку для реактора  $R_2$  ( $v_{s9} = 1$ ), а потім для реактора  $R_1$  ( $v_{s4} = 2$ ).

Але синтезований таким чином алгоритм керування не здатний компенсувати збурення, які завжди присутні в реальних технологічних процесах. Так, наприклад, якщо виникає затримка закінчення дозування речовини  $A$  в реактор  $R_2$  (позиція  $S_{11}$ ) на 5 одиниць часу, то одночасно з її закінченням знову маркується позиція  $S_2$ , що є недопустимим режимом

роботи. Для того щоб гарантувати компенсацію неконтрольованих збурень і забезпечити коректне використання дозаторів не тільки у власному стані, додатково до матриці зворотного зв'язку  $\mathbf{K}_1$  вводяться два елементи, які дозволяють вирішити цю проблему.

Так, позиція  $S_2$  в наступному циклі буде тільки тоді маркована, коли позиція  $S_{11}$  в поточному циклі демаркується, тобто маркується позиція  $S_{12}$ .

Звідси:

$$(\mathbf{K}_1)_{1,12} = 0.$$

За допомогою іншої логічної умови керування, а саме: позиція  $S_9$  повинна в наступному циклі маркуватись, коли позиція  $S_4$  демаркована або  $S_5$  маркована. Тоді можна обчислити інший елемент матриці  $\mathbf{K}_1$ :

$$(\mathbf{K}_1)_{3,5} = 0.$$

Матриця динаміки замкнутої ДБС матиме вигляд:

$$\mathbf{M}_s = (\mathbf{A}_0 \oplus \mathbf{BK}_0)^* (\mathbf{A}_1 \oplus \mathbf{BK}_1),$$

яка має незмінне власне число  $\lambda_s = 7$  і незмінний власний вектор  $\mathbf{v}_s$ .

На рис. 10.3 представлено графік перехідного процесу в ДБС для характерних позицій графа синхронізації  $S_2, S_4, S_9, S_{11}$ . На графіку показано, як об'єкт виходить в усталений режим роботи з початкових умов  $\mathbf{x}(0) = 0$ , а також показана реакція алгоритму керування на збурення у вигляді затримки маркування позиції  $S_5$  на 10 умовних одиниць часу. Графік побудовано за даними розрахункової таблиці 10.2 маркувань позицій для 10 циклів спостереження за системою.

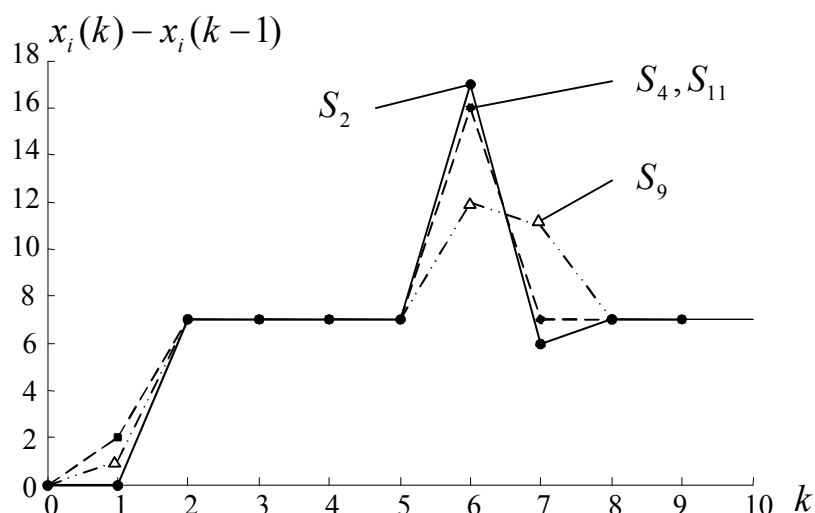


Рисунок 10.3 – Перехідний процес і компенсація збурення в системі

Оскільки граф  $G^C(\mathbf{M}_s)$  має циклічність  $\mu = 1$ , очевидно, що після кожного збурення система знову повертається в усталений стан, обумовлений власним вектором. Дані таблиці 10.2 ілюструють, що дозатори для речовин  $A$  і  $B$  завжди використовуються виключно для одного з вихідних продуктів, тобто положення  $S_2$  і  $S_{11}$  та  $S_4$  і  $S_9$ , ніколи одночасно не маркуються.

Таблиця 10.2 – Часові точки маркування позицій  $S_2, S_4, S_9, S_{11}$

$S \backslash k$	1	2	3	4	5	6	7	8	9	10
$S_2$	0	7	14	21	28	45	51	58	65	72
$S_4$	2	9	16	23	30	46	53	60	67	74
$S_9$	1	8	15	22	29	41	52	59	66	73
$S_{11}$	2	9	16	23	30	46	53	60	67	74

Також необхідно зазначити, що в системі не всі маркери початкового маркування знаходяться в передобластях керованих переходів, а тому необхідною умовою старту алгоритму керування є умова, коли  $x_1(0) \leq x_6(0)$ .



## Висновки

В монографії з загальносистемної точки зору зроблено спробу систематизації, узагальнення та подальшого розвитку досліджень в теорії управління дискретно-безперервними процесами. Основна мета представлених в роботі досліджень – подальший розвиток теорії аналітичного синтезу алгоритмів управління об'єктами дискретно-безперервного класу з застосуванням ідей, що є характерними для сучасної теорії автоматичного управління.

Показано, що один із різновидів ідемпотентної алгебри, а саме  $\text{Max-Plus}$  алгебра, дозволяє отримувати моделі ДБС в формі аналітичних рівнянь в просторі стану з урахуванням особливостей представлення динаміки дискретно-безперервних явищ.

В систематизованому стилі наведено основні поняття і визначення цього математичного апарату. На простих прикладах проілюстровано типові дії та операції, що є базовими для скалярних та векторно-матричних обчислень. Більш детально проаналізовано характерні рівняння простору станів та сформульовано необхідні умови їх розв'язання. Особливу увагу приділено практичним алгоритмам розв'язку класичного спектрального рівняння, яке має широке застосування в теорії аналізу складних динамічних систем.

Сформульовано основні засади аналітичного опису ДБС та наведено практику його застосування на прикладах простих технічних систем.

Розглянуто питання зв'язку між формами представлення ДБС за допомогою графів синхронізації мереж Петрі та відповідною аналітичною моделлю в просторі стану.

Сформульовано специфічні поняття для ДБС, такі як: керований із

зовні перехід об'єкта з одного стану в інший та логічні умови виконання такого переходу. По аналогії з безперервними і дискретними системами введено поняття перехідної функції в дискретно-безперервних системах, що наочно характеризує динаміку поведінки процесу.

Сформульовано умови керованості дискретно-безперервних об'єктів.

Розглянуто узагальнений підхід до синтезу алгоритму керування в замкнутих системах зі зворотним зв'язком за допомогою векторів стану дискретно-безперервного об'єкта в поточному і попередньому циклах роботи ДБС. По аналогії з відомими методами синтезу управління за заданими показниками якості для безперервних систем, в монографії запропоновано підходи, що дозволяють сформулювати логічні умови управління дискретно-безперервними об'єктами, які можна трансформувати в матрицю зворотного зв'язку (матрицю регулятора), при цьому застосовуються формальні процедури матричних перетворень.

В монографії знайшли своє відображення специфічні питання. При аналізі динамічних характеристик замкнутої ДБС використовуються підходи, які відомі в теорії класичних систем. Зокрема, рівняння стану замкнутої системи та властивості матриці динаміки системи. Але по відношенню до ДБС ці поняття мають цілий ряд особливостей. Специфічними для ДБС є проблеми стартового управління, інваріантність до завад та умови компенсації детермінованих збурень. Особливості синтезу керування ДБС проілюстровані прикладами практичного застосування одержаних теоретичних положень.

Представлена робота формує основу для подальшого розвитку теорії синтезу оптимального управління дискретно-безперервними процесами, що є актуальною проблемою. В монографії розглянуто аналіз та синтез управління в рамках класичної теорії замкнутих систем керування зі зворотним зв'язком та за заданими показниками якості управління.

## Список літератури

1. Берж, К. Теория графов и ее приложения [Текст] / К. Берж. – М. : Ин. Лит., 1962. – 319с.
2. Бессараб, В. И. Автоматическое управление многоступенчатым водоотливом шахт [Текст] / В. И. Бессараб, Р. В. Федюн // Матеріали міжнародної конференції з управління "Автоматика - 2001", 10 - 14 вересня 2001р., м. Одеса, Україна : в 2-х т. – Одеса : ОГПУ, 2001. - Т.1. - С. 129-130.
3. Бессараб, В. И. Динамическая модель многоступенчатого водоотлива угольных шахт [Текст] / В. И. Бессараб, Р. В. Федюн // Наукові праці ДонДТУ. Серія: Гірнична електромеханіка. Випуск 16. Донецьк. - Донецьк: ДонДТУ, 2000. – С. 19–25.
4. Бессараб, В. И. Моделирование режимов багатоступінчатого водовідливу вугільних шахт з використанням пакету MATLAB [Текст] / В. И. Бессараб, Р. В. Федюн // Наукові праці ДонДТУ. Серія: Гірнична електромеханіка. Випуск 27. Донецьк. – Донецьк : ДонДТУ, 2001. – С. 29-34.
5. Бессараб, В. И. Оптимальное управление многоступенчатым водоотливом шахт [Текст] / В. И. Бессараб, Р. В. Федюн // Труды Одесского политехнического университета: Научный и производственно-практический сборник по техническим и естественным наукам. Выпуск 3(15). 2001 г. – Одесса : ОГПУ, 2001. - С. 198-203.
6. Бессараб, В.И. Принципы моделирования динамических процессов в многоступенчатых водоотливных установках [Текст] / В. И. Бессараб, Р. В. Федюн // Наукові праці ДонДТУ. Серія: Обчислювальна техніка

- та автоматизація. Випуск 12. Донецьк. – Донецьк : ДонДТУ, ТОВ “Лебідь”, 1999. – С. 33–40.
7. Бессараб, В.І. Розробка моделі багатоступінчатої водовідливної установки вугільної шахти [Текст] / В. І. Бессараб, Р. В. Федюн // Контроль і управління в складних системах (КУСС-2001). Тези доповідей шостої науково-технічної конференції м. Вінниця, 8-12 жовтня 2001 року. – Вінниця : УНІВЕРСУМ-Вінниця, 2001. - С. 204.
  8. Бессараб, В.И. Синтез цифровых систем управления многоступенчатыми водоотливными установками угольных шахт [Текст] / В. И. Бессараб, Р. В. Федюн // Наукові праці ДонДТУ. Серія: Обчислювальна техніка та автоматизація. Випуск 25. Донецьк. – Донецьк : ДонДТУ, 2001. – С. 15–21.
  9. Бессараб, В. И. Применение современных телекоммуникационных технологий в системах комплексной автоматизации угольных шахт [Текст] / В. И. Бессараб, Р. В. Федюн, Г. В. Курносков // Наукові праці ДонНТУ. Серія: Обчислювальна техніка та автоматизація. Випуск 74. Донецьк. – Донецьк: ДонНТУ, 2004. – С. 19–25
  10. Бессараб, В. И. MPL система управления комплексом водоотлива угольных шахт [Текст] / В. И. Бессараб // Материалы XIII Международной конференции по автоматическому управлению “Автоматика -2007”, г. Севастополь, 10-14 сентября 2007 року. Ч.1. – Севастополь : СНУЯЭиП, 2007. – С. 117-119.
  11. Бессараб, В. І. Використання апарата «MAX-PLUS» алгебри для моделювання динаміки в інформаційних мережах із простою топологією [Текст] / В. І. Бессараб, Є.Г. Коваленко // Наукові праці Інституту проблем моделювання в енергетиці ім. Г.Є. Пухова. Випуск 44. – Київ, 2007. – С. 59 – 62.

12. Бессараб, В. И. Модель шахтного водоотлива как комплексного дискретно-непрерывного объекта [Текст] / В. И. Бессараб, В. В. Червинский, Н. В. Червинская // Автоматика 2008: доклады XV міжнародної конференції з автоматичного управління, 23-26 вересня 2008 р. – Одеса : ОНМА, 2008. – С. 652-654
13. Вулих, Б. З. Введение в теорию полуупорядоченных пространств [Текст] / Б. З. Вулих. – М. : Физматгиз, 1961. – 407с.
14. Коэн, Г. Алгебраические средства оценивания характеристик дискретно-событийных систем [Текст] / Г. Коэн, П. Моллер, Ж.-П. Кадра, М. Вьо // ТИИЭР. – 1989. – Т. 77, N 1. – С. 30-53.
15. Кривулин, Н. К. Несмещенная градиентная оценка при оптимизации дискретных динамических систем [Текст] / Н. К. Кривулин // Вычислительная техника и проблемы кибернетики, под ред. Чиркова М.К. Санкт-Петербургский Государственный Университет. Т. 28. – Санкт-Петербург, 1995. – С. 199-219.
16. Кристофидес, Н. Теория графов. Алгоритмический подход [Текст] / Н. Кристофидес. – М. : Мир, 1978. – 432с.
17. Литвинов, Г. Л. Идемпотентный функциональный анализ: алгебраический подход [Текст] / Г. Л. Литвинов, В. П. Маслов, Г. Б. Шпиз // Мат. заметки. – 2001. – Т. 69, №5. – С. 696–799.
18. Литвинов, Г. Л. Линейные функционалы на идемпотентных пространствах. Алгебраический подход [Текст] / Г. Л. Литвинов, В. П. Маслов, Г. Б. Шпиз // Доклады Академии Наук. – 1998. – Т.363, №3. – С. 298-300.
19. Литвинов, Г. Л. Тензорные произведения идемпотентных полумодулей. Алгебраический подход [Текст] / Г. Л. Литвинов, В. П. Маслов, Г. Б. Шпиз // Матем. Заметки. – 1999. – Т. 65, №4. – С.572–585.

20. Маслов, В. П. Идемпотентный анализ и его применение в оптимальном управлении [Текст] / В. П. Маслов, В. Н. Колокольцов. – М.: Физматлит, 1994. – 144с.
21. Милов Д.С. Анализ среднего времени работы систем с очередями на основе алгебраических моделей [Текст] / Д. С. Милов // Сборник тезисов Международной конференции «Математические методы и компьютеры в экономике» г. Пенза, май 1998. – Пенза: Пензенский технологический институт, 1998. – Ч. I. – С. 48-49.
22. Романовский И.В. Асимптотическое поведение дискретного детерминированного процесса с непрерывным множеством состояний [Текст] / И. В. Романовский // Оптимальное планирование. Сборник трудов ИМ СО АН СССР. – 1967. – Т.8. – С. 171-193.
23. Свами, М. Графы, сети и алгоритмы [Текст] : пер. с англ./ М. Свами, К. Тхуласираман; ред. В.А. Горбатого. – М. : Мир, 1984. – 455 с.
24. Татт, У.Т. Теория графов [Текст] / У. Т. Татт; пер с англ. Г.П. Гаврилова. – М. : Мир, 1988. – 424с.
25. Уилсон, Р. Введение в теорию графов [Текст] : пер. с англ. / Р. Уилсон; ред. Г.П. Гаврилова. – М. : Мир, 1977. – 207с.
26. Харари, Ф. Теория графов [Текст] : пер. с англ. / Ф. Харари. – М. : Мир, 1973. – 386с.
27. A linear-system-theoretic view of discrete-event processes and its use for performance evaluation in manufacturing [Text] / G. Cohen, D. Dubois, J. P. Quadrat, M. Viot // IEEE Transactions on Automatic Control, Mar. 1985. – Vol. AC-30, no. 3. – P. 210–220.
28. Abel, D. Petri-Netze Für Ingenieure [Text] / D. Abel. – Berlin, Heidelberg, New York: Springer-Verlag. – 1990. – ISBN 3-540-51814-2
29. Akian, M. Discrete max-plus spectral theory [Text] / M. Akian, S. Gaubert, C. Walsh.// Idempotent Mathematics and Mathematical Physics,

- Contemporary Mathematics; In G. L. Litvinov and V. P. Maslov editors. – American Mathematical Society, 2005. – P. 53–78.
30. Algebraic tools for the performance evaluation of discrete event systems [Text] / G. Cohen, P. Moller, J.-P. Quadrat, M. Viot // IEEE Proceedings. – 1989. – Vol. 77. – P. 39-58.
31. Baccelli, F. Estimates of Cycle Times in Stochastic Petri Nets [Text] / F. Baccelli, P. Konstantopoulos // Lecture Notes in Control and Information Sciences. Vol. 177. – Springer-Verlag, 1992. – P. 1-21.
32. Brandin, B. A. Supervisory control of timed discrete-event systems [Text] / B. A. Brandin, W. M. Wonham // Transactions on Automatic Control. Vol. 39. – 1994. – N2. – P. 329-342.
33. Cassandras, C. G. Introduction to Discrete Event Systems [Text] / C. G. Cassandras S. Lafortune. – Kluwer Academic Publishers, 1999.
34. Cohen, G. Duality and separation theorems in idempotent semimodules [Text] / G. Cohen, S. Gaubert, J.-P. Quadrat // Linear Algebra and Appl. Vol. 379. – 2004. – P.395–422.
35. Cohen, G. Max-plus algebra and system theory: where we are and where to go now [Text] / G. Cohen, S. Gaubert, J.-P. Quadrat // *Annual Reviews in Control* (Elsevier--IFAC). – 1999. – Vol. 23, No. 1. – P. 207-219.
36. Cofer, D. D. A Max-algebra solution to the supervisory control problem for real-time discrete-event systems [Text] / D. D. Cofer, V. K. Garg // Proceedings of the 11<sup>th</sup> International Conference on Analysis and Optimization of Systems, Discrete Event Systems. – Sophia-Antipolis: Springer-Verlag. – 1994. – P. 283-289.
37. Cunningham-Green, R. Minimax Algebra [Text] / R. Cunningham-Green // Lecture Notes in Economics and Mathematical Systems. Vol. 166. – Springer-Verlag, Berlin-Hoidelberg-New York, 1979. – 258p.

38. Cunningham-Green, R. Minimax Algebra and applications [Text] / R. Cunningham-Green // Advanced in Imaging and Electron Physics. Vol. 90. – New York: Academic Press, 1995. – P. 1-121.
39. David, R. Petri Nets and Grafcet: Tools for Modelling Discrete Event Systems [Text] / R. David, H. Alla. – London: Prentice Hall, 1992. – ISBN: 0-13-327537-X.
40. De Schutter, B. A method to find all solutions of a system of multivariate polynomial equalities and inequalities in the max algebra [Text] / B. De Schutter, B. De Moor // Discrete Event Dynamic Systems: Theory and Applications. Vol. 6, no. 2. – Mar. 1996. – P. 115-138.
41. De Schutter, B. Max-Algebraic System Theory for Discrete Event Systems [Text] : PhD thesis / B. De Schutter. – Faculty of Applied Sciences, K.U.Leuven, Leuven, Belgium, Feb. 1996. – 331 p.
42. De Schutter, B. Minimal realization in the max algebra is an extended linear complementarity problem [Text] / B. De Schutter, B. De Moor // Systems & Control Letters. Vol. 25, no. 2. – May 1995. P. 103-111.
43. De Schutter, B. Model predictive control for max-plus-linear discrete event Systems [Text] / B. De Schutter, T. van den Boom // Automatica. Vol. 37. – 2001. – P. 1049-1056.
44. Del Moral, P. Maslov optimization theory. Topological aspects [Text] / P. Del Moral // Idempotency. Publ. of the Newton Institute. – Cambridge: Cambridge Univ. Press, 1998. – P. 354-382.
45. Discrete abstractions of hybrid systems [Text] / R. Alur, T. Henzinger, G. Lafferriere, G. Pappas // Proceedings of the IEEE, 2000. – V. 88. - № 7. – P. 971-984.
46. Gaubert, S. Methods And Applications of (max,+) Linear Algebra. [Text] / S. Gaubert // Technical report, INRIA, Rocquencourt, France, Jan. 1997. – 24 p.



47. Geert Jan Olsder. Eigenvalues of Dynamic Max-Min Systems [Text] / Geert Jan Olsder // Department of Technical Mathematics and Informatics, Delft University of Technology. Discrete Event Dynamic Systems: Theory and Applications 1. – 1991. – P.177-207.
48. Hanisch, H.-M. Petri-Netze in der Verfahrenstechnik [Text] / H.-M. Hanisch. – München, Wien: Oldenbourg-Verlag. – 1992. – ISBN 3-486-22190-6.
49. Kingman, J. F. C. Subadditive ergodic theory [Text] / J. F. C. Kingman // Ann. Probability 1. – 1973. – P. 883-909.
50. König, R. Petri-Netze in der Steuerungs- und Digitaltechnik [Text] / R. König, L. Quäck. – München, Wien: Oldenbourg-Verlag. – 1988. – ISBN 3-486-20735-0.
51. Krivulin, N. K. Using max-algebra linear models in the representation of queueing systems [Text] / N. K. Krivulin // In Proceedings of the 5th SIAM Conference on Applied Linear Algebra, Snowbird, Utah, June 1994. – P. 155–160.
52. Krogh, B.H. Synthesis of feedback control logic for discrete manufacturing systems [Text] / B. H. Krogh , L. E. Holloway // Automatica (Journal of IFAC).Vol.27, n.4. – July 1991. – P.641-651.
53. Litvinov, G. L. Idempotent Interval Analysis and Optimization Problems [Text] / G. L. Litvinov, A. N. Sobolevski // M. V. Lomonosov Moscow State University. The work supported by the RFBR grant No. 99–01–01198 and the Erwin Schrödinger Institute for Mathematical Physics. – January 2001. – 28 p.
54. Mairesse J. A Graphical Representation for Matrices in the (Max,+) Algebra [Text] / J. Mairesse // Architectures paralleles, bases de donnees, reseaux et systemes distribues. Rapport de recherche N 2078. – November 1993. – 36p.

55. Max-Plus Algebra and Applications to System Theory and Optimal Control [Text] / Max-Plus Working Group consisting of M. Akian, G. Cohen, S. Gaubert, J.-P. Quadrat, M. Viot INRIA-Rocquencourt // Proceedings of the International Congress of Mathematicians, Zurich, Switzerland. – August 1994.
56. Merlet, G. Cycle time of stochastic max-plus linear systems [Text] / G. Merlet // Electronic Journal of Probability. Vol. 13. – 2008. – P. 322–340.
57. Mossig, K. Algebraischer Steuerungsentwurf für eine Klasse ereignisdiskreter Prozesse mittels der Max-Plus-Algebra [Text] / K. Mossig. – VDI-Fortschrittsberichte, volume 20. – Düsseldorf: VDI Verlag. – 1996. – 137s.
58. Mossig, K. Einfuehrung in die “Max-Plus”-Algebra zur Beschreibung ereignisdiskreter dynamischer Prozesse [Text] / K. Mossig, A. Rehkopf // Automatisierungstechnik. Vol. 44. – Karlsruhe, 1996. – P. 3-9.
59. Mossig, K. Control design for timed petri nets with the max-plus algebra [Text] / K. Mossig, V. Krebs // Proc. First Asian Control Conference. Vol.2. – 1994. – P. 889-892.
60. Mossig, K. Steuerungsentwurf fuer ereignidiskrete Systeme durch Vorgabe eines Eigenvektors der “Max-Plus”-Algebra [Text] / K. Mossig // Automatisierungstechnik. Vol. 44. – Karlsruhe, 1996. – P. 80-86.
61. Murata, T. Petri Nets: Properties, analysis and application [Text] / T. Murata // Proceedings of the IEEE 77. – 1989. – N4. – P. 541-580.
62. Olsder, G. J. Descriptor systems in the max-min algebra [Text] / G. J. Olsder // Proceedings of the 1st European Control Conference. – Herm`es, Paris. – 1991. – P. 1825-1830.
63. Olsder, G. J. Eigenvalues of dynamic max-min systems [Text] / G. J. Olsder // Discrete Event Dynamic Systems: Theory And Applications. Vol. 1. – Sept. 1991. – P. 177-207.

64. Peterson, J. Petri net theory and the modeling of systems [Text] / J. Peterson. – Prentice Hall. – 1981. – ISBN 0–13–661983–5.
65. Raisch, J. Discrete Abstractions of Continuous Systems — an Input/Output Point of View [Text] / J. Raisch // Mathematical and Computer Modelling of Dynamical Systems. Special issue on Discrete Event Models of Continuous Systems, 2000. - V. 6. - № 1. P. 6-29.
66. Ramadge, P. J. Modular feedback logic for discrete event systems [Text] / P. J. Ramadge, W. M. Wonham // SIAM Journal of Control and Optimization. Vol. 25. – 1987. – N5. – P. 1202-1218.
67. Ramadge, P. J. Supervisory control of a class of discrete even systems [Text] / P. J. Ramadge, W. M. Wonham // SIAM Journal of Control and Optimization. Vol. 25. – 1987. – N5. – P. 206-230.
68. Rehkopf, A. Steuerung des ereignisdiskreten Fertigungsablaufs in einer CIM-Zelle unter Einbeziehung heuristischer und analztischer Optimierungsstrategien [Text] : Dissertation, Prüfungsdatum 04.12.1992 / A. Rehkopf. – Universität Karlsruhe. – 1992. – 144s.
69. Subiono. Power algorithms for (max,+)- and bipartite (min, max, +)-systems [Text] / Subiono, J. van der Woude // Discrete Event Dynamic Systems: Theory and Applications. – 2000. – N 10(4). – P. 369-389.
70. Synchronization and Linearity: An algebra for discrete events systems [Text] / F. Baccelli, G. Cohen, G. J. Olsder, J.-P. Quadrat. – New York : Wiley, 1992.

## Додаток – Лістинг програм моделювання

Лістинг 1 – Виробнича система з паралельно-послідовною обробкою виробів (рис. 6.1).

```
clear
disp(' ');
disp('chapter 6.1');

t1 = 5;
t2 = 4;
t3 = 3;
t4 = 3;
t5 = 2;
t6 = 1;
t7 = 0;

d1 = 6;
d2 = 4;
d3 = 7;
d4 = 2;

A0 = ones(4,4) * -inf;
A0(4,1) = mp_multi(t4,d1);
A0(4,2) = mp_multi(t5,d2);
A0(4,3) = mp_multi(t6,d3);

A1 = ones(4,4) * -inf;
A1(1,1) = d1;
A1(2,2) = d2;
A1(3,3) = d3;
A1(4,4) = d4;

B0 = ones(4,1) * -inf;
B0(1) = t1;
B0(2) = t2;
B0(3) = t3;

A = mp_multi(mp_star(A0),A1)
B = mp_multi(mp_star(A0),B0)

C = ones(1,4) * -inf;
C(1,4) = mp_multi(t7,d4);
C

x0 = ones(4,1) * -inf;

disp('u=0=const');
u = 0;

X(:,1) = mp_sum(mp_multi(A,x0), mp_multi(B,u));
Y(1) = mp_multi(C,X(:,1));
for i=2:10
```

```

        X(:,i) = mp_sum(mp_multi(A,X(:,i-1)), mp_multi(B,u));
        Y(i) = mp_multi(C,X(:,i));
end
X
Y

figure(1);
mp_ganttx(X(:,1:10), [d1 d2 d3 d4]', [0,80]);

[lambda, d] = mp_mcm(A)

clear all

```

**Лістинг 2 – Виробнича система зі змішаною послідовністю обробки виробів (рис. 6.3).**

```

clear
disp(' ');
disp('chapter 6.2');

t1 = 5;
t2 = 4;
t3 = 3;
t4 = 3;
t5 = 2;
t6 = 1;
t7 = 1;

d1 = 6;
d2 = 4;
d3 = 2;
d4 = 7;

A0 = mp_zeros(4,4);
A0(3,1) = mp_multi(t3,d1);
A0(3,2) = mp_multi(t4,d2);
A0(4,2) = mp_multi(t5,d2);
A0(4,3) = mp_multi(t6,d3);

A1 = mp_zeros(4,4);
A1(1,1) = d1;
A1(2,2) = d2;
A1(3,3) = d3;
A1(4,4) = d4;

B0 = mp_zeros(4,1);
B0(1) = t1;
B0(2) = t2;

A = mp_multi(mp_star(A0),A1)
B = mp_multi(mp_star(A0),B0)

C = mp_zeros(1,4);
C(1,4) = mp_multi(t7,d4);
C

```

```
x0 = ones(4,1) * -inf;

disp('u=0=const');
u = 0;

X(:,1) = mp_sum(mp_multi(A,x0), mp_multi(B,u));
Y(1) = mp_multi(C,X(:,1));
for i=2:10
    X(:,i) = mp_sum(mp_multi(A,X(:,i-1)), mp_multi(B,u));
    Y(i) = mp_multi(C,X(:,i));
end
X
Y

figure(1);
mp_gantt(X(:,1:10), [d1 d2 d3 d4]', [0 90]);

[lambda, d] = mp_mcm(A)

clear all
```

Лістинг 3 – Багаторівнева виробнича система (рис. 6.5).

```
clear
disp(' ')
disp('chapter 6.3')

d1 = 1;
d2 = 2;
d3 = 3;
d4 = 4;
d5 = 5;
d6 = 6;
d7 = 5;
d8 = 4;
d9 = 3;
d10 = 2;
d11 = 1;

A0 = mp_zeros(11,11);
A0(2,1) = d1;
A0(4,1) = d1;
A0(4,3) = d3;
A0(5,2) = d2;
A0(5,4) = d4;
A0(6,5) = d5;
A0(7,3) = d3;
A0(8,4) = d4;
A0(8,7) = d7;
A0(9,6) = d6;
A0(9,8) = d8;
A0(10,8) = d8;
A0(11,9) = d9;
A0(11,10) = d10;
A0
```

```

A1 = mp_zeros(11,11);
A1(1,2) = d2;
A1(1,10) = d10;
A1(2,5) = d5;
A1(3,6) = d6;
A1(3,7) = d7;
A1(6,11) = d11;
A1(7,9) = d9;
A1(10,11) = d11;
A1
A1

A = mp_multi(mp_star(A0),A1)

B0 = mp_zeros(11,8);
B0(1,1) = mp_ones;
B0(1,6) = mp_ones;
B0(2,7) = mp_ones;
B0(3,2) = mp_ones;
B0(3,5) = mp_ones;
B0(6,8) = mp_ones;
B0(7,3) = mp_ones;
B0(10,4) = mp_ones;
B0
B0

B = mp_multi(mp_star(A0),B0)

[lambda,iter] = mp_mcm(A)

x = mp_zeros(11,1);
u = mp_ones(8,1);

X(:,1) = mp_sum(mp_multi(A,x),mp_multi(B,u));
for i=2:10
    X(:,i)=mp_sum(mp_multi(A,X(:,i-1)),mp_multi(B,u));
end

X

figure(1);
MPX_RsV = [1 2 0 0;3 4 5 6;7 8 9 0;10 11 0 0];
mp_ganttr(X(:,1:10), [d1 d2 d3 d4 d5 d6 d7 d8 d9 d10 d11]',MPX_RsV, [0
190]);

D=mp_zeros(8,8);
D(1,1)=d2;
D(2,2)=d6;
D(3,3)=d9;
D(4,4)=d11;
D(5,5)=d7;
D(6,6)=d10;
D(7,7)=d5;
D(8,8)=d11;

for i=1:10
    XV(:,i)=[X(2,i);X(6,i);X(9,i);X(11,i);X(7,i);X(10,i);X(5,i);X(11,i)];
    Y=mp_multi(D,XV);
end
Y

```

Лістинг 4 – Обчислення циклічності, елементарних складових та повної частотної проекції матриці  $A$ .

```
clear
disp(' ');
disp('chapter 7, examples 7.1 - 7.4');

A = [mp_one mp_one mp_zero mp_zero; -1 -2 mp_zero mp_zero; mp_zero -1 -1
mp_one; mp_zero mp_zero mp_one mp_one]
[lambda,d] = mp_mcm(A)
x0 = [mp_one; mp_zero; mp_zero; mp_zero]
v = mp_egvl(A,x0)

for i = 1:4
    for j = 1:4
        if(mp_multi(A(i,j),v(j)) == v(i))
            S(i,j) = A(i,j);
        else S(i,j) = mp_zero;
        end;
    end;
end;
S

x1 = [ mp_zero; mp_one; mp_zero; mp_zero]
y = mp_egvl(A,x1)

for i = 1:4
    A_plus(i,1) = v(i);
    A_plus(i,2) = y(i);
end;
A_plus(:,3) = [mp_zero;mp_zero;mp_one;mp_one];
A_plus(:,4) = [mp_zero;mp_zero;mp_one;mp_one];
A_plus

Q1 = mp_multi(A_plus(:,1),A_plus(1,:))
Q34 = mp_multi(A_plus(:,3),A_plus(3,:))
Q = mp_sum(Q1,Q34)

for i = 2:4
    disp(['A',num2str(i)])
    mp_power(A,i)
end;
```

Лістинг 5 – Аналіз динаміки ДБС за допомогою “Мах-plus” алгебри (рис. 8.3)

```
clear
disp(' ')
disp('chapter 8.3')

a = 0; % a = 2;

A = mp_zeros(6,6);
A(1,3) = 2;
A(1,4) = 7-a;
```



```

A(2,1) = 2;
A(2,5) = 6;
A(3,2) = 1+a;
A(3,6) = 7;
A(4,1) = 2;
A(4,5) = 6;
A(5,2) = 1+a;
A(5,6) = 7;
A(6,3) = 2;
A(6,4) = 7-a;
A

TA0 = mp_zeros(6,6);
TA1 = mp_zeros(6,6);
TA0(1,1) = mp_one; TA0(3,3) = mp_one; TA0(5,5) = mp_one;
TA1(2,2) = mp_one; TA1(4,4) = mp_one; TA1(6,6) = mp_one;

A0 = mp_multi(A,TA0)
A1 = mp_multi(A,TA1)

A0_star = mp_star(A0)
M = mp_multi(A0_star,A1)

[lambda,d] = mp_mcm(M)

x0 = mp_ones(6,1);
% x0(6,1) = 2;
v = mp_egv1(M,x0)

X(:,1) = x0;
for k=2:10
    X(:,k) = mp_multi(M,X(:,k-1));
end;
X

Y(:,1) = x0;
for k=2:9
    Y(:,k) = X(:,k)-X(:,k-1);
end;
Y

figure(1);
for k=1:8
    line([k,k+1],[Y(:,k),Y(:,k+1)],'LineWidth',1,'Color','black')
end;

```

### Лістинг 6 – Загальний підхід до синтезу управління в ДБС (рис. 9.2)

```

clear
disp('')
disp('chapter 9.1')

A = mp_zeros(9,9);
A(1,2) = 1; A(1,3) = 1; A(2,6) = 1;
A(2,9) = 1; A(3,1) = 1; A(3,7) = 1;
A(4,2) = 1; A(4,3) = 1; A(5,4) = 1;
A(6,5) = 1; A(7,8) = 1; A(8,6) = 1;
A(8,9) = 1; A(9,1) = 1; A(9,7) = 1;
A

```

```
B = mp_zeros(9,3);
B(2,3) = mp_one; B(5,1) = mp_one; B(7,2) = mp_one; B(8,3) = mp_one;
B

TA0 = mp_zeros(9,9);
TA1 = mp_zeros(9,9);
TA0(1,1) = mp_one; TA0(2,2) = mp_one; TA0(4,4) = mp_one;
TA0(5,5) = mp_one; TA0(7,7) = mp_one; TA0(8,8) = mp_one;
TA1(3,3) = mp_one; TA1(6,6) = mp_one; TA1(9,9) = mp_one;

A0 = mp_multi(A,TA0)
A1 = mp_multi(A,TA1)

A0_star = mp_star(A0)
M = mp_multi(A0_star,A1)

[lambda,d] = mp_mcm(M)

x0 = mp_ones(9,1);
v = mp_egv1(M,x0)
v1 = v - min(v)

KH = mp_zeros(3,9);
KH(1,3) = 1;

BKH = mp_multi(B,KH)

KH(1,3) = mp_zero;
KH(2,5) = 1;

BKH = mp_multi(B,KH)

K0 = mp_zeros(3,9);
K0(1,3) = 1;
K0

A0BK0 = mp_sum(A0,mp_multi(B,K0))
A0BK0_star = mp_star(A0BK0)

K1 = mp_zeros(3,9);
K1(3,9) = 3;
K1

A1BK1 = mp_sum(A1,mp_multi(B,K1))

Ms = mp_multi(A0BK0_star,A1BK1)
[lambda_s,d_s] = mp_mcm(Ms)
```

**Лістинг 7 – Приклад реалізації управління в ДБС з жорсткими логічними умовами (рис. 9.3).**

```
clear
disp('')
disp('chapter 9.4')

A = mp_zeros(6,6);
A(1,2) = 1; A(2,1) = 1; A(2,3) = 1;
```

```

A(3,4) = 1; A(3,6) = 1; A(4,1) = 1;
A(4,3) = 1; A(5,4) = 1; A(5,6) = 1;
A(6,5) = 1;
A

B = mp_zeros(6,2);
B(2,1) = mp_one; B(3,2) = mp_one; B(4,1) = mp_one; B(5,2) = mp_one;
B

TA0 = mp_zeros(6,6);
TA1 = mp_zeros(6,6);
TA0(1,1) = mp_one; TA0(4,4) = mp_one; TA0(5,5) = mp_one;
TA1(2,2) = mp_one; TA1(3,3) = mp_one; TA1(6,6) = mp_one;

A0 = mp_multi(A,TA0)
A1 = mp_multi(A,TA1)

A0_star = mp_star(A0)
M = mp_multi(A0_star,A1)

[lambda,d] = mp_mcm(M)

x0 = mp_ones(6,1);
v = mp_egv1(M,x0)
v1 = v - 2

X(:,1) = v1;
for k=2:10
    X(:,k) = mp_multi(M,X(:,k-1));
end;
X

Y(:,1) = x0;
for k=2:9
    Y(:,k) = X(:,k)-X(:,k-1);
end;
Y

figure(1);
for k=1:8
    line([k,k+1],[Y(:,k),Y(:,k+1)],'LineWidth',1,'Color','black')
end;

K0 = mp_zeros(2,6);
% K0(2,1) = 10;

A0BK0 = mp_sum(A0,mp_multi(B,K0))
A0BK0_star = mp_star(A0BK0)

K1 = mp_zeros(2,6);
K1(2,1) = 10;

A1BK1 = mp_sum(A1,mp_multi(B,K1))

clear v v1

Ms = mp_multi(A0BK0_star,A1BK1)
[lambda_s,d_s] = mp_mcm(Ms)
v = mp_egv1(Ms,x0)
v1 = v - v(2)

```

Лістинг 8 – Приклад, що ілюструє компенсацію збурень в ДБС.

```
clear
disp('')
disp('chapter 9.6.2')

Ms = mp_zeros(6,6);
Ms(1,2) = 1; Ms(2,2) = 2; Ms(2,3) = 1;
Ms(3,2) = 7; Ms(3,3) = 2; Ms(3,6) = 1;
Ms(4,2) = 2; Ms(4,3) = 1; Ms(5,2) = 7;
Ms(5,3) = 2; Ms(5,6) = 1; Ms(6,2) = 8;
Ms(6,3) = 3; Ms(6,6) = 2;
Ms

[lambda_s,d_s] = mp_mcm(Ms)

x0 = mp_ones(6,1);
v = mp_egv1(Ms,x0)

X(:,1) = x0;
for k=2:10
    X(:,k) = mp_multi(Ms,X(:,k-1));
end;
X

Y(:,1) = x0;
for k=2:9
    Y(:,k) = X(:,k)-X(:,k-1);
end;
Y

figure(1);
for k=1:8
    line([k,k+1],[Y(:,k),Y(:,k+1)],'LineWidth',1,'Color','black')
end;

clear all
```

Лістинг 9 – Приклад синтезу системи керування хімічним реактором змішування компонентів (глава 10).

```
clear
disp(' ')
disp('chapter 10')

disp('МАТРИЦА M1')

AF = mp_zeros(5,5);
AF(2,1) = mp_one;
AF(3,2) = 1;
AF(4,3) = mp_one;
AF(5,4) = 1;
AF(1,5) = 4;
```

```

AF

TAF0 = mp_zeros(5,5);
TAF1 = mp_zeros(5,5);
TAF0(2,2) = mp_one; TAF0(3,3) = mp_one; TAF0(4,4) = mp_one;
TAF0(5,5) = mp_one; TAF1(1,1) = mp_one;
AF0 = mp_multi(AF,TAF0)
AF1 = mp_multi(AF,TAF1)

AF0_star = mp_star(AF0)
MF = mp_multi(AF0_star,AF1)

[lambdaF,dF] = mp_mcm(MF)

xF0 = mp_ones(5,1);
vF = mp_egv1(MF,xF0)

disp('МАТРИЦА M2')

AS = mp_zeros(7,7);
AS(2,1) = mp_one;
AS(3,2) = 1;
AS(4,3) = mp_one;
AS(5,4) = 1;
AS(6,5) = mp_one;
AS(7,6) = 1;
AS(1,7) = 4;
AS

TAS0 = mp_zeros(7,7);
TAS1 = mp_zeros(7,7);
TAS0(2,2) = mp_one; TAS0(3,3) = mp_one; TAS0(4,4) = mp_one;
TAS0(5,5) = mp_one; TAS0(6,6) = mp_one; TAS0(7,7) = mp_one;
TAS1(1,1) = mp_one;
AS0 = mp_multi(AS,TAS0)
AS1 = mp_multi(AS,TAS1)

AS0_star = mp_star(AS0)
MS = mp_multi(AS0_star,AS1)

[lambdaS,dS] = mp_mcm(MS)

xS0 = mp_ones(7,1);
vS = mp_egv1(MS,xS0)

disp('МАТРИЦА M')

M = mp_zeros(12,12);
for i=1:5
    for j=1:5
        M(i,j)=MF(i,j);
    end;
end;
for i=6:12
    for j=6:12
        M(i,j)=MS(i-5,j-5);
    end;
end;
M

```

```

A0 = mp_zeros(12,12);
for i=1:5
    for j=1:5
        A0(i,j)=AF0(i,j);
    end;
end;
for i=6:12
    for j=6:12
        A0(i,j)=AS0(i-5,j-5);
    end;
end;
A0

A1 = mp_zeros(12,12);
for i=1:5
    for j=1:5
        A1(i,j)=AF1(i,j);
    end;
end;
for i=6:12
    for j=6:12
        A1(i,j)=AS1(i-5,j-5);
    end;
end;
A1

B = mp_zeros(12,4);
B(2,1)=mp_one;
B(4,2)=mp_one;
B(9,3)=mp_one;
B(11,4)=mp_one;
B

K0 = mp_zeros(4,12);
K0(2,10) = mp_one;
K0(4,3) = mp_one;
K0

A0BK0 = mp_sum(A0,mp_multi(B,K0))
A0BK0_star = mp_star(A0BK0)

K1 = mp_zeros(4,12);
K1(3,5) = mp_one;
K1(1,12) = mp_one;
K1

A1BK1 = mp_sum(A1,mp_multi(B,K1))

Ms = mp_multi(A0BK0_star,A1)
[lambda_s,d_s] = mp_mcm(Ms)

xMs0 = mp_ones(12,1);
vMs = mp_egv1(Ms,xMs0)

X(:,1) = xMs0;
% for k=2:11
%     X(:,k) = mp_multi(Ms,X(:,k-1));
%     if (k==4)
%         X(5,k) = X(5,k) + 10;
%     end;

```

```
% end;

for k = 2:6
    X(:,k) = mp_multi(Ms,X(:,k-1));
end;
X(:,6)

X(5,6) = X(5,6) + 10;
X(1,6) = X(1,6) + 10;
X(12,6) = X(12,6) + 10;

for k = 7:11
    X(:,k) = mp_multi(Ms,X(:,k-1));
end;
X(:,6)
X

Y(:,1) = xMs0;
for k = 2:11
    Y(:,k) = X(:,k)-X(:,k-1);
end;
Y

figure(1);
for k=1:10
    line([k,k+1],[Y(2,k),Y(2,k+1)],'LineWidth',1,'Color','black')
    line([k,k+1],[Y(4,k),Y(4,k+1)],'LineWidth',1,'Color','black')
    line([k,k+1],[Y(9,k),Y(9,k+1)],'LineWidth',1,'Color','black')
    line([k,k+1],[Y(11,k),Y(11,k+1)],'LineWidth',1,'Color','black')
end;
```