

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
Кафедра Програмної інженерії

АТЕСТАЦІЙНА РОБОТА **Пояснювальна записка**

рівень вищої освіти другий (магістерський)

Дослідження каскадної композиції для аналізу дескрипторних систем

Виконав:

студент 2 курсу, групи ІПЗм-18-4

Приходько К.Д.

спеціальності 121 – Інженерія програмного
забезпечення

Тип програми: освітньо-наукова

Освітня програма Інженерія програмного
забезпечення

Керівник:

проф. Руткас А.Г

Допускається до захисту

Зав. кафедри, проф

Дудар З.В.

2020 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук

Кафедра Програмної інженерії

Рівень вищої освіти другий (магістерський)

Спеціальність 121 – Інженерія програмного забезпечення

(код і повна назва)

Тип програми: освітньо-наукова

Освітня програма: Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

« ____ » _____ 20 ____ р.

ЗАВДАННЯ
НА АТЕСТАЦІЙНУ РОБОТУ

студентові Приходьку Кирилу Дмитровичу

1. Тема роботи Дослідження каскадної композиції для аналізу дескрипторних систем

затверджена наказом університету від 27.03.2020р. № 473Ст

2. Термін подання студентом роботи до екзаменаційної комісії “ ____ ” _____ 2020 р.

3. Вихідні дані до роботи електронні ресурси за обраною тематикою, вимоги до функціональності програми, методи обчислення дескрипторних систем, платформа розробки C++, середовище розробки Visual Studio 20174. Перелік питань, що потрібно опрацювати в роботі мета роботи, аналіз проблемної галузі і постановка задачі, дослідження методів обчислення дескрипторних систем, методи оптимізації алгоритму

5. Консультанти розділів роботи

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Спецчастина	проф. Руткас А.Г.		21.05.2020

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1.	Аналіз проблемної області	27.01.20 – 10.02.20	виконано
2.	Постановка задачі	11.02.20 – 13.02.20	виконано
3.	Дослідження існуючих методів та моделей	14.02.20 – 04.03.20	виконано
4.	Розробка математичної моделі та алгоритму	04.03.20 – 14.03.20	виконано
5.	Програмна реалізація	15.03.20 – 25.03.20	виконано
6.	Перевірка розробленого алгоритму	26.03.20 – 30.03.20	виконано
7.	Підготовка пояснювальної записки	01.04.20 – 01.05.20	виконано
8.	Підготовка презентації та доповіді	02.05.20 – 08.05.20	виконано
9.	Попередній захист	12.05.20	виконано
10.	Нормоконтроль, рецензування	13.05.20 – 22.05.20	виконано
11.	Занесення диплома в електронний архів	22.05.20 – 25.05.20	виконано
12.	Допуск до захисту у зав. кафедри	25.05.20	виконано

Дата видачі завдання “___” _____ 2020 р.

Студент _____ Приходько К.Д.
(підпис)

Керівник роботи _____ проф. Руткас А.Г.
(підпис)

РЕФЕРАТ

Атестаційна робота магістра містить: 60 стор., 12 рис., 14 джерел.

C++, ДЕСКРИПТОРНА СИСТЕМА КЕРУВАННЯ, РЕГУЛЯРНИЙ ЖМУТОК МАТРИЦЬ, ПОСЛІДОВНА КОМПОЗИЦІЯ, ЕКВІВАЛЕНТНІ СИСТЕМИ.

Метою роботи є дослідження послідовної композиції дескрипторних систем керування, розробка та реалізація алгоритму, що допоможе автоматично розраховувати коефіцієнти каскадно(послідовно) скомпонованої системи за допомогою коефіцієнтів вихідних систем за умови заданих ресурсних обмежень.

Засобами і методами розробки є Visual Studio 2017, C++.

Результатом розробки є програма «MatrixComposer», яка може використовуватися користувачами для розрахунку коефіцієнтів каскадно(послідовно) скомпонованої дескрипторної системи.

ABSTRACT

The explanatory note for the certification of the master's degree includes 60 pages, 12 figures, 14 sources.

C++, DESCRIPTOR CONTROLLED SYSTEM, REGULAR MATRIX PENCIL, SEQUENTIAL COMPOSITION, DECOMPOSITION, EQUIVALENT SYSTEMS.

The purpose of the work is to study the sequential composition of descriptor control systems, development and implementation of an algorithm that will help to automatically calculate the coefficients of cascaded (sequentially) composed system using the coefficients of derived systems under given resource constraints.

The tools and methods of development are Visual Studio 2017, C ++.

The result is a “MatrixComposer” software that can be used by users to calculate the coefficients of the cascade (sequentially) composed descriptor system.

ЗМІСТ

Вступ.....	7
1. Аналіз проблемної області та постановка задачі.....	8
1.1 Аналіз предметної галузі.....	8
1.2 Теорія керування.....	9
1.3 Поняття композиції.....	11
1.4 Постановка задачі.....	12
2. Дослідження альтернативного методу композиції.....	16
2.1. Загальні відомості.....	16
2.2. Розв’язання рівняння станів.....	19
2.3. Теореми та зауваження.....	21
2.4. Послідовна композиція дескрипторних систем.....	21
3. Опис та програмування алгоритму.....	28
3.1. Опис алгоритму.....	28
3.2. Реалізація алгоритму.....	32
Висновки.....	39
Перелік використаних джерел.....	40
Додаток А Слайди презентації.....	42
Додаток Б Лістинг коду.....	48

ВСТУП

Дескрипторні системи з'являються у багатьох сферах, таких як системи електроживлення, електричні мережі, аерокосмічна інженерія, хімічні процеси, соціально-економічні системи, мережевий аналіз, біологічні системи, аналіз часових рядів тощо.

Теорія лінійних дескрипторних систем керування є важливою частиною в загальній галузі управління теорії систем і привернула багато уваги в останні два десятиліття. Незважаючи на те, що теорія лінійних дескрипторних систем керування є дуже насиченою за змістом, існує лише кілька вичерпних книг на цю тему, наприклад [3]. Існують деякі інші книги та докторські наукові тези, що стосуються дескрипторних систем, але всі вони зосереджені на дуже особливих темах.

Метою роботи є аналіз різноманітних джерел, з'ясування та консолідація інформації щодо досліджень послідовної композиції дескрипторних систем керування та реалізація алгоритму, що допоможе автоматично розраховувати коефіцієнти скомпонованої системи за допомогою коефіцієнтів вихідних систем.

У цій роботі буде розглянуто кілька джерел та зроблені попередні висновки щодо досліджень послідовної композиції дескрипторних систем керування а також реалізовано та запрограмовано алгоритм, що допоможе автоматично розраховувати коефіцієнти скомпонованої дескрипторної системи за допомогою коефіцієнтів компонентних систем при послідовній (каскадній) композиції.

В ході атестаційної роботи магістра було розроблено презентацію(див. додаток А). Лістинг програмної реалізації наведено в додатку Б.

1 АНАЛІЗ ПРОБЛЕМНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Аналіз предметної галузі

Система керування – це систематизований набір засобів впливу на підконтрольний об'єкт для досягнення цим об'єктом певної мети.

Об'єктом системи керування можуть бути як технічні об'єкти так і люди. Об'єкт системи керування може складатися з інших об'єктів, які можуть мати постійну структуру взаємозв'язків.

Технічна система керування – це пристрій або набір пристроїв для маніпулювання поведінкою інших пристроїв або систем.

Об'єктом керування може бути будь-яка динамічна система чи її модель. Стан об'єкта характеризується деякими кількісними величинами, що змінюються в часі, тобто змінними стану.

В технологічних процесах у ролі таких змінних може виступати температура, густина, в'язкість сировини або механічні переміщення (кутові або лінійні) і їхня швидкість, електричні змінні, температура технологічного обладнання, у бізнес-процесах – це курс цінних паперів тощо.

Аналіз і синтез систем керування проводиться методами спеціального розділу математики – теорії керування.

Системи керування поділяють на два великі класи:

- Автоматизовані системи управління (АСУ) – працюють за участю людини в контурі керування;
- Системи автоматичного керування (САК) – працюють без участі людини в контурі керування.

Дескрипторні неперервні і дискретні системи виникають при дослідженні динаміки складних керованих об'єктів механіки, електротехніки, економіки тощо. Рівняння руху неперервних дескрипторних систем не розв'язані відносно старших похідних, тобто є диференціально-алгебраїчними. Це обумовлює складність аналітичного дослідження проблем стійкості й керування для таких систем. Відомі методи побудови та дослідження розв'язків класу лінійних дескрипторних систем базуються на застосуванні теорії канонічних форм матричних жмуктів та узагальнених обернених матриць.

Існує велика кількість літератури про дескрипторні системи управління, див., наприклад, і бібліографію в цих роботах. Дескрипторна система управління, яка виникає при моделюванні перехідних режимів радіотехнічних ланцюгів, досліджується в роботі [4]. В теорії диференціальних ігор вивчаються дескрипторні системи керування з двома конфліктними управліннями [5]. Стани дескрипторних систем управління також можуть описуватися диференціальними рівняннями другого порядку, не розв'язаними щодо старшої похідної. Ці дескрипторні системи служать гарними скінченновимірними апроксимаціями нескінченновимірних систем управління типу Соболева [6].

1.2 Теорія керування

Теорія керування (англ. Control theory) – наука про принципи і методи керування різними системами, процесами і об'єктами.

Теоретичним підґрунтям теорії керування є кібернетика і теорія інформації.

Суть теорії керування полягає в тому, що на основі системного аналізу об'єкта керування проектується його математична модель, після чого

розробляється та реалізується алгоритм керування для отримання бажаних характеристик плинності процесу або досягнення цілей керування.

Дана галузь знань добре розвинена і знаходить широке застосування в сучасній техніці у вигляді такої дисципліни, як теорія автоматичного керування.

У соціально-економічних системах теорію керування присвячено способам і методам аналізу, прогнозуванню і можливостям регулювання діяльності різних людських спільнот. Теорія керування, як і будь-яка наука, має свою методологію і методичне забезпечення. Проте в області природничих наук і техніки, теорія керування має значно більше успіхів, ніж в соціально-економічній сфері, де, очевидно, діє обмеження, що випливає з принципу, – «система не може пояснити саму себе».

Представлення простору стану лінійних систем є основоположним для аналізу та проектування динамічних систем[10]. Сучасна теорія управління значною мірою спирається на представлення простору стану динамічних систем, що полегшує характеристику властивостей динамічних систем.

З моменту введення поняття стану, вивчення лінійних систем у самому представленні простору стану сформувалося як постійно активний напрямок досліджень, що охоплює широкий спектр тем від основних понять стійкості, керованості, спостережливості, надмірності та мінімальності до більш заплутаних властивостей скінченних і нескінченних нульових структур та геометричних підпросторів.

Поглиблює розуміння лінійних систем розвиток сучасної теорії управління. Вимогливі очікування від теорії управління нав'язують постійно зростаючий попит на розуміння та використання більш тонких властивостей лінійних систем.

Важливість теорії лінійних систем та активної дослідницької діяльності, пов'язаної з нею, відображаються також у постійній публікації підручників та монографій з теорії лінійних систем, а також публікація класичних праць Калмана та його колег[1], Гілберта[11] та інших.

Існує два типи циклів керування: відкритий цикл керування і замкнутий цикл керування (із зворотнім зв'язком). Теорія керування переймається лише системами із замкненим циклом керування. Розглянемо приклад такої системи з життя.

У випадку із замкненим циклом керування, алгоритм керування залежить від зворотного зв'язку між учасниками процесу. Гарним прикладом можна вважати автоматичний водонагрівач(або бойлер), який при замкненому циклі використовує для порівняння температури води всередині бака з бажаною температурою(заданою користувачем) спеціальний пристрій – термостат. Такий підхід дозволяє задати управління регулятора, що підтримувати бажану температуру всередині бака вмикаючи та вимикаючи сам пристрій нагрівання.

Тобто регулятор із замкненим циклом забезпечує зворотній зв'язок, який дозволяє контрольному пристрою керувати змінами процесу нагрівання і саме тому такі контролери із замкненим циклом керування ще називають контролерами зі зворотним зв'язком.

1.3 Поняття композиції

Композиція у програмуванні або ж Об'єктна композиція, також Агрегація та включення - це створення об'єктів існуючих класів як елементів інших класів. Про композицію також часто говорять як про *«відношення-приналежності»* за принципом *«у машини має корпус, колеса і двигун»*.

Вкладені об'єкти нового класу зазвичай оголошуються закритими, що робить їх недоступними для прикладних програмістів, що працюють з класом. З іншого боку, творець класу може змінювати ці об'єкти, не порушуючи роботи існуючого клієнтського коду. Крім того, заміна вкладених об'єктів на стадії виконання програми дозволяє динамічно змінювати її поведінку.

Механізм наслідування такої гнучкості не має, оскільки для вихідних класів встановлюються обмеження, що перевіряються на стадії компіляції.

На відміну від наслідування, в композиції тип відносин є *Has-a* тобто має (машина має двигун). В наслідуванні ж тип відносин між породженим об'єктом і батьківським є *Is-a* зв'язком, тобто якщо об'єкт кішка породжено від тварина, то кішка є тварина (*cat is a pet*).

Різниця між агрегацією і композицією полягає в тому, що у зв'язку композиція життя об'єкта контролюється його "володарем", а під час агрегації – не контролюється.

Композиція (суперпозиція) функцій (відображень) в математиці – функція, побудована з двох функцій таким чином, що результат першої функції є аргументом другої.

Композиція функцій $f: X \rightarrow Y$ та $g: Y \rightarrow Z$ будується так: аргумент x з X застосовується до першої функції f , а її результат y з Y застосовується як аргумент до другої функції g .

1.4 Постановка задачі

З сучасної теорії управління добре відомо, що двома основними математичними зображеннями динамічних систем є матриця передачі та рівняння для внутрішнього стану системи. Перше зображення описує лише властивості процесу перетворення входу у вихід системи, тоді як друге дає подальше розуміння структурної властивості системи. У цьому розділі ми опишемо моделі дескрипторних систем.

Для отримання моделі простору станів практичної системи нам потрібно вибрати деякі фізичні змінні, такі як швидкість, вага, температура або прискорення, яких достатньо для характеристики системи. Потім фізичними відносинами серед змінних або за допомогою деяких методів ідентифікації

моделі – набір рівнянь можна встановити. Природно, що цей набір рівнянь зазвичай є диференціальними та алгебраїчними рівняннями, які утворюють математичну модель системи.

Правильно визначенні вектори стану $x(t)$ та вхідного вектора $u(t)$, які утворені фізичними змінними системи та вихідним вектором $y(t)$, елементи яких належним чином вибрані вимірні змінні системи, цей набір рівнянь можна організувати на два рівняння: одне - це так зване рівняння стану, яке є наступним у загальній формі:

$$f(x'(t), x(t), u(t), t) = 0,$$

а інше – вихідне рівняння, або рівняння спостереження, яке знаходиться в формі:

$$g(x(t), u(t), v(t), t) = 0,$$

де f і g - векторні функції відповідних розмірів відносно $x(t)$, $x'(t)$, $u(t)$, $y(t)$ та t . Для дескрипторних систем рівняння спостереження має вигляд:

$$g(x'(t), x(t), u(t), v(t), t) = 0.$$

Наведені вище рівняння дають математичну модель загальної нелінійної динамічної системи керування з неперервним часом.

Надалі для системи керування Φ ми використовуватимемо скорочене позначення $\Phi = \Phi(u(t), x(t), v(t))$, де $u(t)$ керування(вхід), $x(t)$ – стан, $v(t)$ – вихід системи, де $0 < t < T$.

Для кожного моменту t значення вектор-функцій $u(t)$, $x(t)$, $v(t)$ належать лінійним просторам входів U , станів X , виходів V відповідно. Зараз буде описано композицію складових систем $\Phi_j(u_j, x_j, v_j)$ із просторами входів U_j , станів X_j , виходів V_j відповідно, $j = 1, 2$.

Послідовною композицією систем Φ_1, Φ_2 називається система

$$\Phi = \Phi_1 \otimes \Phi_2,$$

отримана ототожненням виходу v_1 системи Φ_1 зі входом u_2 системи Φ_2 , вхід u системи Φ збігається з входом u_1 системи Φ_1 , вихід v системи Φ збігається з виходом v_2 системи Φ_2 .

Щодо просторів входів U_1, U_2, U і виходів V_1, V_2, V систем Φ_1, Φ_2, Φ припускаємо, що

$$V_1 = U_2, U = U_1, V = V_2.$$

Простори станів X і образів Y системи Φ є прямі суми відповідних просторів систем Φ_1, Φ_2 :

$$X = X_1 + X_2, Y = Y_1 + Y_2.$$

На рис. 1 зображено принцип послідовної композиції $\Phi = \Phi_1 \otimes \Phi_2$.

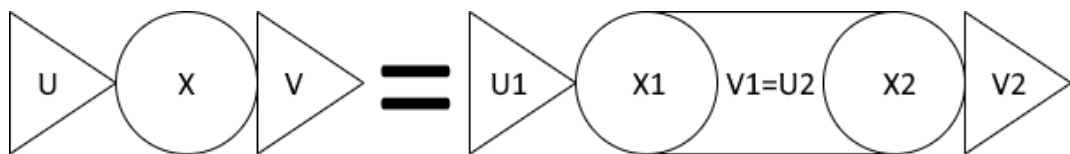


Рисунок 1 – Послідовна композиція.

Метою роботи є реалізація та програмування алгоритму, що допоможе автоматично розраховувати коефіцієнти рівнянь (4), (5) скомпонованої

дескрипторної системи Φ за допомогою відповідних коефіцієнтів складових систем Φ_1, Φ_2 при послідовній(каскадній) композиції.

Інакше кажучи, функції та рівняння опису

$$\begin{aligned} f(x'(t), x(t), u(t)) &= 0, \\ g(x'(t), x(t), u(t), v(t)) &= 0, \end{aligned}$$

системи Φ слід побудувати за допомогою функцій та рівнянь опису

$$\begin{aligned} f_j(x'_j(t), x_j(t), u_j(t)) &= 0, \\ g_j(x'_j(t), x_j(t), u_j(t), v_j(t)) &= 0, \end{aligned}$$

складових систем $\Phi_j, j = 1, 2$.

Сформуємо стислу специфікацію до розробляємої програми:

- Програма має працювати на основі алгоритмів, описаних у цій роботі.
- Програма має мати зручний алгоритм вводу даних та простий і прозорий вигляд результатів. Алгоритм має бути універсальним незалежно від розмірів вхідних даних.
- Програма повинна бути оптимізована для роботи з великими обсягами обчислень.
- Програма має бути підтримувана на операційних системах від Windows XP до Windows 10 та оптимізована для слабких машин.
- Програма має відловлювати помилки та обробляти їх(наприклад, повідомляти користувача про неправильні вхідні дані).

2 ДОСЛІДЖЕННЯ АЛЬТЕРНАТИВНОГО МЕТОДУ КОМПОЗИЦІЇ

2.1 Загальні відомості

У цьому розділі ми дослідимо альтернативний метод каскадної композиції дескрипторних систем[7].

Лінійні системи керування зі входом(керуванням) $u(t)$, станом $x(t)$, виходом $v(t)$ і початковим станом x_0 у класичній монографії Калмана та Фалба[1] описуються відношеннями

$$\frac{dx}{dt} = ax(t) + bu(t), v(t) = cx(t) + du(t), 0 \leq t \leq T, \quad (1)$$

$$x(0) = x_0. \quad (2)$$

У випадку стаціонарної системи, коли матричні коефіцієнти a, b, c, d не залежать від часу, при переході до амплітуд періодичних коливань чи перетворенні Лапласа $\hat{u}(\lambda), \hat{x}(\lambda), \hat{v}(\lambda)$ функцій $u(t), x(t), v(t)$ що задовільняють (1) з нульовим початковим станом $x_0 = 0$ (2), отримуємо

$$\hat{x}(\lambda) = (\lambda E - a)^{-1}b\hat{u}(\lambda), \hat{v}(\lambda) = [d + c(\lambda E - a)^{-1}b]\hat{u}(\lambda). \quad (3)$$

При переході до перетворень Лапласа функції доповнюємо нулем для $t > T$. Тут і надалі символ E означає єдиничну матрицю відповідної розмірності. Із виразу (3) бачимо, що система (1), (2) не має резонансу(особливості) при нескінченній частоті, амплітуда стану $\hat{x}(\lambda)$ зменшується до нуля при необмеженому зростанні комплексної частоти λ . Остання властивість може не виконуватися у деяких реальних системах, наприклад у наведених нижче прикладах радіотехнічних фільтрів.

Тому розглянемо більш загальне рівняння для стану x та виходу v , не розв'язані відносно похідної стану:

$$\frac{d}{dt}(Ax) + Bx(t) = Fu(t), 0 \leq t \leq T, \quad (4)$$

$$v(t) = \frac{d}{dt}(Mx) + Nx + Ku, 0 \leq t \leq T, \quad (5)$$

де $A, B \in R^{m \times m}$, $M, N \in R^{p \times m}$, $F \in R^{m \times n}$, $K \in R^{p \times n}$. Множина дійсних $m \times n$ -матриць визначається $R^{m \times n}$.

Системи керування з рівняннями станів виду (4) у теорії керування називають дескрипторними[2, 3]. У теорії диференціальних ігор вивчаються дескрипторні системи з двома конфліктними керуваннями[4]. Стани дескрипторних систем керування також можуть описуватися диференціальними рівняннями другого порядку, не розв'язаними відносно старшої похідної. Ці дескрипторні системи служать гарними скінченновимірними апроксимаціями нескінченновимірних систем керування типу Соболева[6].

Відношення (4), (5), (2) визначають систему $\Phi = \Phi(u, x_0, x, v)$. Рівняння (4) і систему Φ назвемо регулярними, якщо їх характеристичний жмуток матриць $\lambda A + B$ є регулярним, тобто $\chi(\lambda) = \det(\lambda A + B) \neq 0$ (многочлен $\chi(\lambda)$ – не тотожний нуль), виродженими, якщо $\det A = 0$ (або матриця A незворотна) та явними, якщо $A = E$. Відповідно до введеної термінології система (1), (2) є явною з нульовою матрицею $M = 0$. Далі будемо розглядати лише регулярні дескрипторні системи керування.

За операторного трактування системи Φ коефіцієнти в (4), (5) будемо представляти у якості лінійних операторів $A, B : X \rightarrow Y$, $F : U \rightarrow Y$, $M, N : X \rightarrow V$, $K : U \rightarrow V$, що діють у дійсних скінченновимірних евклідових просторах X, Y, U, V .

Тоді матричні коефіцієнти інтерпретуються як матриці відповідних операторів відносно визначених базисів.

Тут $\dim X = \dim Y = m$, $\dim U = n$, $\dim V = p$.

Зокрема, можна вибрати $X = Y = R^m$, $U = R^n$, $V = R^p$. Значення функцій входів(керувань) $u(t)$ належать простору входів(керувань) U , функції станів $x(t)$ – простору станів X , функцій виходів $v(t)$ – простору виходів V , функцій $Fu(t)$, $Ax(t)$, $Bx(t)$ – простору образів Y . Регулярність характеристичного жмутка операторів $(\lambda A + B) : X \rightarrow Y$ та, відповідно, дескрипторної системи Φ еквівалентна тому, що знайдеться хоча б одне дійсне число λ_0 , при якому існує зворотний оператор $(\lambda_0 A + B)^{-1} : Y \rightarrow X$, визначений на всьому просторі Y .

У цій роботі досліджуються питання композиції дескрипторних систем керування. Розглядається послідовне з'єднання двох або більше систем керування в одну. Ці дослідження відносяться до систем управління найбільш загального виду – до дескрипторних систем керування.

Наводяться умови, які дозволяють уявити систему як послідовне з'єднання систем менших розмірностей. У якості прикладу у роботі демонструються отримані результати на прикладі дескрипторних систем, що описують перехідні процеси у радіотехнічних фільтрах. Побудована композиція має наглядну фізичну інтерпретацію у вигляді класичного каскадного з'єднання чотирьохполюсників.

Визначимо $C^k([0, T], X)$ клас вектор-функцій $x(t) : [0, T] \rightarrow X$, які k разів безперервно диференційовані за усіх $t \in [0, T]$; $C^0 = C$. Для канонічного базиса евклідового m -вимірного простору R^m використовуємо визначення $\{\vec{e}_k\}_{k=1}^m$. При розгляданні операторів у просторах типу R^m матриці операторів запишемо відносно цих базисів. Визначимо $\text{Lin}\{\dots\}$ лінійну оболонку векторів, $\text{Ker } K$ – ядро оператора K , $\dim X$ – розмірність простору X .

2.2 Розв'язання рівняння станів

Опишемо стани системи Φ , тобто розв'язки рівняння

$$\frac{d}{dt}(Ax) + Bx(t) = Fu(t), 0 \leq t \leq T,$$

якими названо такі функції $x(t) \in C([0, T], X)$, що $Ax(t) \in C^1([0, T], Y)$ і задовільняється рівняння (4).

В часному випадку явної системи

$$\frac{dx}{dt} = ax(t) + bu(t), v(t) = cx(t) + du(t), 0 \leq t \leq T, \quad x(0) = x_0,$$

з неперервним керуванням $u(t)$ та початковою умовою (2) існує єдине $x(t)$, яке допускає зображення у вигляді формули Коші чи формули варіації сталих. При цьому вектор x_0 в (2) може бути будь яким вектором із простору станів X . Для виродженого рівняння (4) може не існувати розв'язань $x(t)$ з визначеними початковими даними [8], і виникає задача опису множини допустимих початкових даних.

Зазвичай, розв'язання виродженого рівняння (4) здійснюється за допомогою методу Вейерштрасса приведення регулярного жмутка матриць $\lambda A + B$ до канонічної форми [8].

Цей метод складно формалізується та не веде до загальної формули розв'язання $x(t)$ типу формули Коші. У зв'язку з цим, використаємо інший функціонально-аналітичний метод побудови явних формул для розв'язання та опису початкового многовиду рівняння (4).

Метод, що розглядається, засновано на використанні двох пар проєкційних матриць P_1, P_2 та Q_1, Q_2 , які відповідають спектральним проєкторам типу Рісса у просторі станів X і в просторі образів Y системи[12]:

$$P_1 = \frac{1}{2\pi i} \oint_{|\lambda|=\gamma} (\lambda A + B)^{-1} A d\lambda, \quad P_2 = E - P_1, \quad (6)$$

$$Q_1 = \frac{1}{2\pi i} \oint_{|\lambda|=\gamma} A(\lambda A + B)^{-1} d\lambda, \quad Q_2 = E - Q_1,$$

тут γ – радіус кола у комплексній площині з центром у нулі, що охоплює увесь скінченний спектр жмутка матриць $\lambda A + B$ так, що для всіх кінечних власних чисел λ_k виконується умова $|\lambda_k| < \gamma$. Практичне інтегрування за комплексною змінною $\lambda = \gamma e^{i\varphi}$ ($0 \leq \varphi \leq 2\pi$) виконується окремо для кожного елементу матриць $(\lambda A + B)^{-1} A$ та $A(\lambda A + B)^{-1}$ з використанням теореми про лишки. Матриці P_k, Q_k дійсні, якщо дійсні матриці A, B .

Визначимо матриці

$$G = AP_1 + BP_2 = Q_1A + Q_2B, \quad H = G^{-1}Q_2A, \quad S = -G^{-1}Q_1B, \quad (7)$$

за допомогою яких зручно записується явна формула для розв'язання $x(t)$ початкової задачі (4), (2). Матриця H є нильпотентною з індексом нильпотентності ν . Число ν співпадає з індексом $r = \text{ind}(A, B)$ жмутка матриць $\lambda A + B$, коли $\det A = 0$. Якщо $\det A \neq 0$ ($\exists A^{-1}$), то $r = 0$.

Для визначення індексу жмутка матриць також використовують ступінну оцінку резольвенти $(\lambda A + B)^{-1}$ в околі нескінченно віддаленої точки $|\lambda| \geq \gamma$:

$$\|(\lambda A + B)^{-1}\| \leq C_0 |\lambda|^{k-1}, 0 \leq k \leq m = \dim X, \quad (8)$$

яка впливає з формули обернення $m \times m$ -матриці. В цій формулі використовується операторна норма, яка є коректною після комплексифікації просторів X, Y та переходу до комплексних розширень операторів A, B [6]. Індекс $r = \text{ind}(A, B)$ жмутка матриць $\lambda A + B$ є мінімальне з цілих чисел k , що допускають оцінку (8).

Наведемо теореми, описані у використаному методі. У теоремі встановлюються умови узгодження початкового стану x_0 дескрипторної системи та управління $u(t)$, які забезпечують однозначну розв'язуваність початкового завдання (4), (2). Також наводиться явна формула, яка описує відповідний стан системи – розв'язання задачі Коші (4), (2).

2.3 Теореми та зауваження

Теорема 1[9]

Нехай $\lambda A + B$ – регулярний жмуток $m \times m$ -матриць з індексом $r = \nu$, компонента $Q_1 F u(t)$ правої частини рівняння (4) неперервна на $[0, T]$, а для компоненти $Q_2 F u(t)$ виконані умови гладкості

$$H^k G^{-1} Q_2 F u(t) \in C^k([0, T], X), k = 0, 1, \dots, \nu - 1, \quad (9)$$

та умови узгодження з початковим вектором

$$P_2 x_0 = \sum_{k=0}^{\nu-1} (-1)^k \frac{d^k}{dt^k} [H^k G^{-1} Q_2 F u(0)]. \quad (10)$$

Тоді задача Коші (4),(2) має єдиний розв'язок, який допускає зображення

$$x(t) = e^{St}P_1x_0 + \int_0^t e^{S(t-\tau)}G^{-1}Q_1Fu(\tau)d\tau + \sum_{k=0}^{\nu-1}(-1)^k \frac{d^k}{dt^k} [H^kG^{-1}Q_2Fu(t)]. \quad (11)$$

Положимо $0^0 = E$, де 0 – квадратна нульова матриця відповідної розмірності.

Зауваження 1.

Якщо система не вироджена (існує обернена матриця A^{-1}), тоді проектори (6) та матриці (7) приймають вигляд $P_1 = Q_1 = E$, $P_2 = Q_2 = 0$, $G = A$, $S = -A^{-1}B$; умову узгодження (10) виконано для будь-якого початкового вектора $x_0 \in X$, у правій частині формули (11) третій доданок дорівнює нулю і рівняння (11) є формула Коші для розв'язання рівняння $x'(t) = Sx(t) + A^{-1}Fu(t)$.

Якщо жмуток $\lambda A + B$ не має кінечних власних чисел, тобто $\det(\lambda A + B) \equiv \text{const} \neq 0$ то $P_1 = 0$, $Q_1 = 0$, $P_2 = E$, $Q_2 = E$, $G = B$, $H = B^{-1}A$ і третій доданок у правій частині формули (11) дає єдиний розв'язок $x(t)$ рівняння (4) з єдиним допустимим початковим вектором $x_0 = P_2x_0$, що визначається за формулою (10).

Зауваження 2.

Нехай в умовах теореми 1 керування неперервне та в умовах (9) вимоги гладкості збільшені на одиницю:

$$u(t) \in C([0, T], U), H^kG^{-1}Q_2Fu(t) \in C^{k+1}([0, T], X), k = 0, 1, \dots, \nu - 1. \quad (12)$$

Тоді стан $x(t)$ (11) є неперервно диференційованим, а вихід $v(t)$ (5) – неперервним. Для зручності перевіримо замість серії вимог (12) одну більш сильну вимогу $Q_2Fu(t) \in C^\nu([0, T], Y)$.

Для амплітуд періодичних коливань $\hat{u}(\lambda), \hat{x}(\lambda), \hat{v}(\lambda)$ з комплексною частотою λ , тобто перетворень Лапласа функцій $u(t), x(t), v(t)$, введемо частотні матриці-функції $\Theta(\lambda)$ та $w(\lambda)$ або відображення входу на стан та входу на вихід:

$$\hat{x}(\lambda) = \Theta(\lambda)\hat{u}(\lambda), \quad \hat{v}(\lambda) = w(\lambda)\hat{u}(\lambda)$$

Для явної системи (1) внаслідок (3) отримаємо

$$\Theta(\lambda) = (\lambda E - a)^{-1}b, \quad w(\lambda) = d + c(\lambda E - a)^{-1}b.$$

Для дескрипторної системи Φ (4), (5) з нульовим початковим станом $x_0 = 0$ (2) маємо

$$\hat{x}(\lambda) = (\lambda A + B)^{-1}F\hat{u}(\lambda), \quad \hat{v}(\lambda) = [K + (\lambda M + N)(\lambda A + B)^{-1}F]\hat{u}(\lambda),$$

якщо λ не є власним числом характеристичного жмутка системи, тобто існує резольвента $(\lambda A + B)^{-1}$. Отже, для дескрипторної системи Φ визначимо частотні матриці-функції $\Theta(\lambda)$ та $w(\lambda)$ як

$$\Theta(\lambda) = (\lambda A + B)^{-1}F, \tag{13}$$

$$w(\lambda) = K + (\lambda M + N)(\lambda A + B)^{-1}F = K + (\lambda M + N)\Theta(\lambda).$$

Тут $w(\lambda)$ – передаточна матриця-функція дескрипторної системи. За операторної трактовки системи Φ вводимо до розгляду частотні оператори функції $\Theta(\lambda)$ та $w(\lambda)$ у комплексних оболонках вихідних просторів та замінюємо вихідні оператори на їх комплексні розширення.

2.4 Послідовна композиція дескрипторних систем

Принцип послідовної композиції (послідовного з'єднання) двох систем керування $\Phi_j = \Phi_j(u_j, x_{j0}, x_j, v_j)$, $j = 1, 2$, в одну систему керування $\Phi(u, x_0, x, v) = \Phi_1 \otimes \Phi_2$ полягає в тому, що вихід v_1 першої системи ототожнюється зі входом u_2 другої; вхід u результуючої системи Φ співпадає з входом u_1 системи Φ_1 ; вихід v системи Φ – з виходом v_2 системи Φ_2 .

Нехай системи Φ_j описуються співвідношеннями

$$\frac{d}{dt}(A_j x_j) + B_j x_j(t) = F_j u_j(t), \quad (14)$$

$$v_j(t) = \frac{d}{dt}(M_j x_j) + N_j x_j(t) + K_j x_j(t), 0 \leq t \leq T, x_j(0) = x_{j0}.$$

Дійсні евклідові простори входів, станів, образів та виходів системи Φ_j позначаються як U_j, X_j, Y_j та V_j . Відносно просторів входів та виходів цих систем припустимо, що

$$V_1 = U_2, U = U_1, V = V_2. \quad (15)$$

Система Φ має простір входів $U = U_1$ та виходів $V = V_2$. Відносно простору станів X та простору образів Y системи Φ припустимо, що вони є прямими сумами відповідних просторів систем Φ_1, Φ_2 :

$$X = X_1 \dot{+} X_2, Y = Y_1 \dot{+} Y_2. \quad (16)$$

На рис. 2 зображено принцип послідовної композиції $\Phi = \Phi_1 \otimes \Phi_2$.

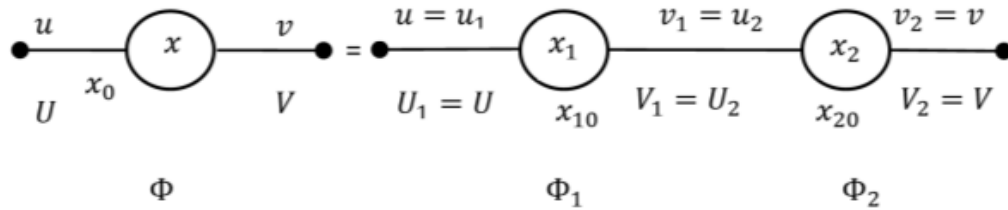


Рисунок 2 – Візуалізація станів систем при послідовній композиції.

Отриману послідовну композицію Φ опишемо в операторній формі (4), (5), (2) у припущенні, що

$$u(t) = u_1(t), x_0 = x_{10} + x_{20}, x(t) = x_1(t) + x_2(t), \quad (17)$$

$$v(t) = v_2(t), u_2(t) = v_1(t).$$

Якщо у відношеннях (14) врахувати рівняння (17), то отримаємо

$$\frac{d}{dt}(A_1 x_1) + B_1 x_1(t) = F_1 u(t), \quad (18)$$

$$\frac{d}{dt}(-F_2 M_1 x_1 + A_2 x_2) - F_2 N_1 x_1(t) + B_2 x_2(t) = F_2 K_1 u(t), \quad (19)$$

$$v(t) = \frac{d}{dt}(K_2 M_1 x_1 + M_2 x_2) + K_2 N_1 x_1(t) + N_2 x_2(t) + K_2 K_1 u(t). \quad (20)$$

Введемо до розгляду оператори $A, B : X \rightarrow Y$, $F : U \rightarrow Y$, $M, N : X \rightarrow V$ та $K : U \rightarrow V$, які представимо в блочній формі згідно з прямими розкладаннями просторів X, Y (16) і рівняннями (15) для просторів U, V :

$$A = \begin{bmatrix} A_1 & 0 \\ -F_2 M_1 & A_2 \end{bmatrix}, \quad B = \begin{bmatrix} B_1 & 0 \\ -F_2 N_1 & B_2 \end{bmatrix}, \quad F = \begin{bmatrix} F_1 \\ F_2 K_1 \end{bmatrix}, \quad (21)$$

$$M = [K_2 M_1 \quad M_2], \quad N = [K_2 N_1 \quad N_2], \quad K = K_2 K_1.$$

Тепер рівняння (18), (19) записуються у вигляді (4), а вихід (20) – у вигляді

$$v(t) = \frac{d}{dt}(Mx) + Nx + Ku, \quad 0 \leq t \leq T.$$

Систему керування Φ (4), (5), (2) з просторами станів X і образів Y (16), з просторами входів U і виходів V (15), з блочними операторними коефіцієнтами (21), відповідним прямим сумам (16), з початковим вектором x_0 , входом $u(t)$ і виходом $v(t)$ в (17) називають послідовним з'єднанням або послідовною композицією $\Phi = \Phi_1 \otimes \Phi_2$ систем Φ_1, Φ_2 (14).

Підпростори X_2, Y_2 утворюють інваріантну пару підпросторів відносно жмутка операторів $\lambda A + B$, бо справедливі включення

$$AX_2 \subset Y_2, \quad BX_2 \subset Y_2. \quad (22)$$

При послідовній композиції дескрипторних систем управління $\Phi = \Phi_1 \otimes \Phi_2$ резольвента характеристичного жмутка системи Φ виражається через резольвенти характеристичних жмутків систем Φ_j для всіх невластних чисел λ цих жмутків:

$$(\lambda A + B)^{-1} = \begin{bmatrix} (\lambda A_1 + B_1)^{-1} & 0 \\ (\lambda A_2 + B_2)^{-1} F_2 (\lambda M_1 + N_1) (\lambda A_1 + B_1)^{-1} & (\lambda A_2 + B_2)^{-1} \end{bmatrix}.$$

Звідси отримуються оцінки резольвенти та індексу $r = \text{ind}(A, B)$ характеристичного жмутка системи Φ , використовуючи індекси $r_j = \text{ind}(A_j, B_j)$ характеристичних жмутків систем Φ_j :

$$\|(\lambda A + B)^{-1}\| \leq C_0 |\lambda|^{r_1+r_2-1} \quad (|\lambda| \gg 1), \quad r \leq r_1 + r_2.$$

Амплітуда періодичних коливань систем Φ, Φ_1, Φ_2 пов'язані рівняннями

$$\hat{u}(\lambda) = \hat{u}_1(\lambda), \hat{u}_2(\lambda) = \hat{v}_1(\lambda), \hat{v}(\lambda) = \hat{v}_2(\lambda), \quad \hat{x}(\lambda) = \hat{x}_1(\lambda) + \hat{x}_2(\lambda).$$

Тому, як для відкритих систем, частотні оператор-функції $\Theta(\lambda), w(\lambda)$ системи Φ допускають зображення за допомогою частотних оператор-функцій $\Theta_1(\lambda), w_1(\lambda)$ та $\Theta_2(\lambda), w_2(\lambda)$ систем Φ_1 та Φ_2 :

$$\Theta(\lambda) = \begin{bmatrix} \Theta_1(\lambda) \\ \Theta_2(\lambda)w_1(\lambda) \end{bmatrix}, w(\lambda) = w_2(\lambda)w_1(\lambda). \quad (23)$$

Тут оператор-функція $\Theta(\lambda) : U \rightarrow X_1 + X_2$ записана у блочній формі відповідно до прямого розкладання. При послідовній композиції $\Phi = \Phi_1 \otimes \Phi_2$ передаточна оператор-функція $w(\lambda)$ допускає мультиплікативне уявлення через передаточні оператор-функції $w_1(\lambda), w_2(\lambda)$.

3 ОПИС ТА ПРОГРАМУВАННЯ АЛГОРИТМУ

3.1 Опис алгоритму

У попередньому розділі нами було розглянуто, досліджено та детально описано метод розкладу дескрипторної системи керування[7], де ми представляємо дескрипторну систему керування як послідовне з'єднання систем менших розмірностей. Такий підхід до представлення системи дозволяє нам займатися як композицією так і декомпозицією таких систем.

Взявши за основу описаний метод, було сформовано алгоритм, який демонструє необхідні перетворення та операції при композиції кількох систем. На вході нам потрібно описати вихідні системи, які буде об'єднано.

Першу систему опишемо у вигляді рівнянь (4), (5) з відповідними коефіцієнтами:

$$\begin{aligned} \frac{d}{dt}(A_1 x_1) + B_1 x_1(t) &= F_1 u_1(t), \\ v_1(t) &= \frac{d}{dt}(M_1 x_1) + N_1 x_1 + K_1 u_1, \\ 0 \leq t \leq T. \end{aligned}$$

Зробимо те саме для другої системи:

$$\begin{aligned} \frac{d}{dt}(A_2 x_2) + B_2 x_2(t) &= F_2 u_2(t), \\ v_2(t) &= \frac{d}{dt}(M_2 x_2) + N_2 x_2 + K_2 u_2, \\ 0 \leq t \leq T. \end{aligned}$$

Зі сформованих систем отримаємо матричні коефіцієнти (які в умовах реальних розрахунків можуть також бути просто скалярами):

- $A_1, B_1, F_1, M_1, N_1, K_1$ з першої системи,
- $A_2, B_2, F_2, M_2, N_2, K_2$ з другої системи.

Користуючись попередньо отриманими виразами, перепишемо наші системи як послідовно з'єднані в операторній формі (4), (5), пам'ятаючи, що згідно припущення

$$u_2(t) = v_1(t).$$

У другій системі замість $u_2(t)$ підставимо $v_1(t)$, та отримаємо

$$\frac{d}{dt}(A_2x_2) + B_2x_2(t) = F_2 \left(\frac{d}{dt}(M_1x_1) + N_1x_1 + K_1u_1 \right).$$

Тепер за правилами перенесемо частину виразу з правої частини до лівої:

$$\begin{aligned} \frac{d}{dt}(A_2x_2 - F_2M_1x_1) - F_2N_1x_1(t) + B_2x_2(t) &= F_2K_1u_1, \\ \frac{d}{dt}(-F_2M_1x_1 + A_2x_2) - F_2N_1x_1(t) + B_2x_2(t) &= F_2K_1u_1, \end{aligned}$$

$$v(t) = \frac{d}{dt}(K_2M_1x_1 + M_2x_2) + K_2N_1x_1(t) + N_2x_2(t) + K_2K_1u(t).$$

Звідси ми отримуємо оператори

$$A, B : X \rightarrow Y, F : U \rightarrow Y, M, N : X \rightarrow V, K : U \rightarrow V,$$

які представляємо в блочній формі згідно з прямим розкладом просторів X , $Y(16)$ і рівняннями для просторів U , $V(15)$:

$$A = \begin{bmatrix} A_1 & 0 \\ -F_2 M_1 & A_2 \end{bmatrix}, \quad B = \begin{bmatrix} B_1 & 0 \\ -F_2 N_1 & B_2 \end{bmatrix}, \quad F = \begin{bmatrix} F_1 \\ F_2 K_1 \end{bmatrix},$$

$$M = [K_2 M_1 \quad M_2], \quad N = [K_2 N_1 \quad N_2], \quad K = K_2 K_1.$$

Тепер ми можемо записати вихід системи $v = v_2$ – у вигляді

$$v(t) = \frac{d}{dt}(Mx) + Nx + Ku, \quad 0 \leq t \leq T.$$

Тобто спрощене завдання алгоритму для зображення композиції полягає в тому щоб, отримавши матричні коефіцієнти вихідних систем $A_1, B_1, F_1, M_1, N_1, K_1, A_2, B_2, F_2, M_2, N_2, K_2$ отримати матричні коефіцієнти A, B, F, M, N, K кінцевої, об'єднаної системи

$$\frac{d}{dt}(Ax) + Bx(t) = Fu(t), \quad 0 \leq t \leq T.$$

Розглянемо приклад послідовної композиції трьох систем.

Композицію зручніше обробляти ітеративно, починаючи з кінця. Для цього спочатку розглянемо приклад розрахунку для системи $\Phi_2 = \Phi_{21} \otimes \Phi_{22}$ з наступними матричними коефіцієнтами:

$$A_{21} = 0, \quad B_{21} = 1, \quad F_{21} = (1 \quad 0), \quad M_{21} = \begin{pmatrix} 0 \\ -C \end{pmatrix},$$

$$N_{21} = \begin{pmatrix} 0 \\ -g \end{pmatrix}, \quad K_{21} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},$$

$$A_{22} = 0, \quad B_{22} = 1, \quad F_{22} = (0 \quad 1), \quad M_{22} = \begin{pmatrix} -L_2 \\ 0 \end{pmatrix},$$

$$N_{22} = \begin{pmatrix} -r_2 \\ 0 \end{pmatrix}, \quad K_{22} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

Підставимо ці коефіцієнти у вищезгадане представлення операторів у блочній формі:

$$\begin{aligned}
 A_2 &= \begin{bmatrix} A_{21} & 0 \\ -F_{22}M_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ -(0 & 1) \begin{pmatrix} 0 \\ -C \end{pmatrix} & 0 \end{bmatrix} = \begin{bmatrix} 0 & C \\ 0 & 0 \end{bmatrix}, \\
 B_2 &= \begin{bmatrix} B_{21} & 0 \\ -F_{22}N_{21} & B_{22} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -(0 & 1) \begin{pmatrix} 0 \\ -g \end{pmatrix} & 1 \end{bmatrix} = \begin{bmatrix} 1 & g \\ 0 & 1 \end{bmatrix}, \\
 F_2 &= \begin{bmatrix} F_{21} \\ F_{22}K_{21} \end{bmatrix} = \begin{bmatrix} (1 & 0) \\ (0 & 1) \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \\
 M_2 &= [K_{22}M_{21} \quad M_{22}] = \left[\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ -C \end{pmatrix} \quad \begin{pmatrix} -L_2 \\ 0 \end{pmatrix} \right] = \begin{bmatrix} -L_2 & 0 \\ 0 & -C \end{bmatrix}, \\
 N_2 &= [K_{22}N_{21} \quad N_{22}] = \left[\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ -g \end{pmatrix} \quad \begin{pmatrix} -r_2 \\ 0 \end{pmatrix} \right] = \begin{bmatrix} -r_2 & 0 \\ 0 & -g \end{bmatrix}, \\
 K_2 &= K_{22}K_{21} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}
 \end{aligned}$$

Тепер розглянемо композицію отриманої системи з ще одною. Для цього розглянемо приклад розрахунку для системи $\Phi = \Phi_1 \otimes \Phi_{21} \otimes \Phi_{22}$ з наступними матричними коефіцієнтами:

$$\begin{aligned}
 A_1 &= 0, \quad B_1 = 1, \quad F_1 = (0 \quad 1), \quad M_1 = \begin{pmatrix} -L_1 \\ 0 \end{pmatrix}, \quad N_1 = \begin{pmatrix} -r_1 \\ 0 \end{pmatrix}, \quad K_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \\
 A_2 &= \begin{pmatrix} 0 & C \\ 0 & 0 \end{pmatrix}, \quad B_2 = \begin{pmatrix} 1 & g \\ 0 & 1 \end{pmatrix}, \quad F_2 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad M_2 = \begin{pmatrix} -L_2 & 0 \\ 0 & -C \end{pmatrix}, \\
 N_2 &= \begin{pmatrix} -r_2 & 0 \\ 0 & -g \end{pmatrix}, \quad K_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.
 \end{aligned}$$

Підставимо ці коефіцієнти у вищезгадане представлення операторів у блочній формі:

$$A = \begin{bmatrix} A_1 & 0 \\ -F_2M_1 & A_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ -\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} -L_1 \\ 0 \end{pmatrix} & \begin{pmatrix} 0 & C \\ 0 & 0 \end{pmatrix} \end{bmatrix} =$$

$$\begin{aligned}
&= \begin{bmatrix} 0 & 0 \\ \begin{pmatrix} 0 \\ L_1 \end{pmatrix} & \begin{pmatrix} 0 & C \\ 0 & 0 \end{pmatrix} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & C \\ L_1 & 0 & 0 \end{bmatrix}, \\
B &= \begin{bmatrix} B_1 & 0 \\ -F_2 N_1 & B_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} -r_1 \\ 0 \end{pmatrix} & \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} g \\ 1 \end{pmatrix} \end{bmatrix} = \\
&= \begin{bmatrix} 1 & 0 \\ \begin{pmatrix} 0 \\ r_1 \end{pmatrix} & \begin{pmatrix} 1 & g \\ 0 & 1 \end{pmatrix} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & g \\ r_1 & 0 & 1 \end{bmatrix}, \\
F &= \begin{bmatrix} F_1 \\ F_2 K_1 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 \end{pmatrix} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}, \\
M &= [K_2 M_1 \quad M_2] = \left[\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} -L_1 \\ 0 \end{pmatrix} \quad \begin{pmatrix} -L_2 & 0 \\ 0 & -C \end{pmatrix} \right] = \begin{bmatrix} -L_1 & -L_2 & 0 \\ 0 & 0 & -C \end{bmatrix}, \\
N &= [K_2 N_1 \quad N_2] = \left[\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} -r_1 \\ 0 \end{pmatrix} \quad \begin{pmatrix} -r_2 & 0 \\ 0 & -g \end{pmatrix} \right] = \begin{bmatrix} -r_1 & -r_2 & 0 \\ 0 & 0 & -g \end{bmatrix}, \\
K &= K_2 K_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}
\end{aligned}$$

3.2 Реалізація алгоритму

Для реалізації алгоритму було обрано мову C++, яка є максимально оптимальною для складних обчислень завдяки прямій взаємодії з пам'яттю. Для зручності завдання вихідних даних було реалізовано за рахунок зміни текстового файлу, а пуск програми обчислення реалізовано bat-файлом.

Роздивимось детальніше деякі ключові моменти реалізації алгоритму.

По-перше, підстановка до представлення операторів у блочній формі потребує множення матриць, тому було розроблено універсальний метод множення матриць будь-якої розмірності. Реалізація цього методу представлена на рисунку 3.


```

std::vector<Telement> multiply(std::vector<Telement> m, std::vector<Telement> n)
{
    int row = sqrt(m.size());
    int col = n.size()/row;
    std::vector<Telement> result;

    for(int i = 0; i < row; i++)
        for(int j = 0; j < col; j++)
        {
            Telement element;

            for(int k = 0; k < row; k++)
            {
                element = element + m[k + row * i] * n[j + col * k];
            }
            result.push_back(element);
        }

    return result;
}

```

Рисунок 3 – Універсальний метод множення матриць.

Метод отримує у якості параметрів дві матриці, записані в особливому типі даних *Telement*. Наступним кроком метод ітеративно перемножує елементи матриць за правилами множення: спочатку йде за строками, далі за стовпчиками до самого кінця підсумкової матриці розміром [кількість строк першої матриці] x [кількість стовпчиків другої матриці].

Реалізація особливого типу даних *Telement*, який забезпечує коректне множення елементів матриць, що можуть бути представлені як скалярами, так і буквенними та комбінованими виразами представлено на рисунку 4.

Даний метод в залежності від того, елемент у нас скаляр чи комбінований вираз для розрахунків використовується або представлення елементу як числовий тип даних з плаваючою комою *float number*, або як асоціативний масив *std::map<std::string, float> var*.

Для кожного представлення прописані інструкції перевизначення операторів складання та множення(необхідні для нашого алгоритму).

```

struct Telement
{
    float number = 0;
    std::map<std::string, float> var;

    Telement operator*(const Telement& other) const
    {
        Telement result = *this;
        if(number == 0 && !var.empty())
        {
            result.number = 1;
        }

        int otherNumber = other.number;
        if(other.number == 0 && !other.var.empty())
        {
            otherNumber = 1;
        }

        result.number *= otherNumber;
        std::string total;
        for(auto & x:other.var)
        {
            if(x.second == 0)
                continue;

            result.number *= x.second;
            total += x.first;
        }

        for(auto & x:var)
        {
            if(x.second == 0)
                continue;

            result.number *= x.second;

            if(other.var.find(x.first) != other.var.end())
            {
                total += x.first;
            }
        }

        result.var.clear();
        if(!total.empty())
        {
            result.var[total] = number;
            result.number = 0;
        }

        return result;
    }

    Telement operator+(const Telement& other) const
    {
        Telement result = *this;

        result.number += other.number;
        for(auto & x:other.var)
        {
            result.var[x.first] += x.second;
        }

        return result;
    }
};

```

Рисунок 4 – Реалізація типу даних *Telement*.

Також розглянемо метод який дозволяє проаналізувати дані, що надані користувачем, та визначити тип елементу: скаляр чи комбінований. Реалізація метода представлена на рисунку 5.

```

std::vector<Telement> parseVector(std::string data){

    std::istringstream parser(data);
    std::vector<Telement> result;
    std::string element;

    while(parser>>element)
    {
        bool isDigit = true;
        for(auto c:element)
        {
            if(c < '0' || c > '9')
            {
                isDigit = false;
                break;
            }
        }
        Telement temp;
        if(isDigit)
        {
            temp.number = std::stof(element);
        }
        else
        {
            temp.var[element] = 1;
        }
        result.push_back(temp);
    }

    return result;
}

```

Рисунок 5 – Реалізація парсера.

На початку ми отримуємо елементи вхідних векторів та матриць та записуємо їх у вигляді *Telement* як змінні $A_2, B_2, F_2, M_2, N_2, K_2, A_1, B_1, F_1, M_1, N_1$ ($K_2 = K_1$). Також оголосимо підсумкові блоки A, B, F, M, N, K .

Кожне представлення операторів у блочній формі розраховується за різними алгоритмами. Розглянемо їх детальніше.

Розрахунок блоку $A = \begin{bmatrix} A_1 & 0 \\ -F_2 M_1 & A_2 \end{bmatrix}$ представлено на рисунку 6.

```

{
  int size = sqrt(A2.size());

  A.resize((size + 1) * (size + 1));
  A[0] = A1[0];
  std::vector<Telement> F2M1 = multiply(F2, M1);
  for(int i = 1; i < size + 1; i++)
  {
    A[i * (size + 1)] = F2M1[i - 1];
    A[i * (size + 1)].number = -A[i * (size + 1)].number;
  }
  for(int i = 0; i < size; i++)
  for(int j = 0; j < size; j++)
  {
    A[j + 1 + (i + 1) * (size + 1)] = A2[j + i * size];
  }
}

```

Рисунок 6 – Розрахунок блоку А.

Розрахунок блоку $B = \begin{bmatrix} B_1 & 0 \\ -F_2 N_1 & B_2 \end{bmatrix}$ представлено на рисунку 7.

```

{
  int size = sqrt(B2.size());

  B.resize((size + 1) * (size + 1));
  B[0] = B1[0];
  std::vector<Telement> F2N1 = multiply(F2, N1);
  for(int i = 1; i < size + 1; i++)
  {
    B[i * (size + 1)] = F2N1[i - 1];
    B[i * (size + 1)].number = -B[i * (size + 1)].number;
  }

  for(int i = 0; i < size; i++)
  for(int j = 0; j < size; j++)
  {
    B[j + 1 + (i + 1) * (size + 1)] = B2[j + i * size];
  }
}

```

Рисунок 7 – Розрахунок блоку В.

Розрахунок блоку $F = \begin{bmatrix} F_1 \\ F_2 K_1 \end{bmatrix}$ представлено на рисунку 8.

```

{
  F.insert(F.end(), F1.begin(), F1.end());

  std::vector<Telement> F2K1 = multiply(F2, K2);
  F.insert(F.end(), F2K1.begin(), F2K1.end());
}

```

Рисунок 8 – Розрахунок блоку F.

Розрахунок блоку $M = [K_2 M_1 \quad M_2]$ представлено на рисунку 9.

```

{
int size = sqrt(K2.size());

std::vector<Telement> K2M1 = multiply(K2, M1);
M.resize(size * (size + 1));

for(int i = 0; i < size; i++)
{
M[i * size] = K2M1[i];
}

size = sqrt(M2.size());

for(int i = 0; i < size; i++)
for(int j = 0; j < size; j++)
{
M[j + 1 + i * (size + 1)] = M2[j + i * size];
}
}

```

Рисунок 9 – Розрахунок блоку M.

Розрахунок блоку $N = [K_2 N_1 \quad N_2]$ представлено на рисунку 10.

```

{
int size = sqrt(K2.size());

std::vector<Telement> K2N1 = multiply(K2, N1);
N.resize(size * (size + 1));
for(int i = 0; i < size; i++)
{
N[i * size] = K2N1[i];
}

size = sqrt(N2.size());

for(int i = 0; i < size; i++)
for(int j = 0; j < size; j++)
{
N[j + 1 + i * (size + 1)] = N2[j + i * size];
}
}

```

Рисунок 10 – Розрахунок блоку N.

Розрахунок блоку $K = K_2 K_1$ представлено на рисунку 11.

```

{
K = multiply(K2, K2);
}

```

Рисунок 11 – Розрахунок блоку K.

Після розрахунків ми зберігаємо отримані блоки, виводимо їх користувачу у консолі та записуємо до файлу результат. Реалізовано це завдяки методу-принтеру, який представлено на рисунку 12.

```
void printVector(std::vector<Telement> data, std::string name, std::ofstream & file)
{
    std::cout<<name<<" = ";
    file<<name<<" = ";

    for(auto x:data)
    {
        if(x.number != 0 || x.var.empty())
        {
            std::cout<<x.number;
            file<<x.number;
        }

        for(auto y:x.var)
        {
            if(y.second != 1)
            {
                std::cout<<y.second;
                file<<y.second;
            }

            if(y.second != 0)
            {
                std::cout<<y.first;
                file<<y.first;
            }
        }

        std::cout<<" ";
        file<<" ";
    }

    std::cout<<"\n";
    file<<"\n";
}
```

Рисунок 12 – Реалізація метода-принтера.

На цьому роботу програми завершено.

ВИСНОВКИ

Дескрипторні системи з'являються у багатьох сферах, таких як системи електроживлення, електричні мережі, аерокосмічна інженерія, хімічні процеси, соціально-економічні системи, мережевий аналіз, біологічні системи, аналіз часових рядів тощо.

Теорія лінійних дескрипторних систем керування є важливою частиною в загальній галузі теорії систем управління і привернула багато уваги в останні два десятиліття. Незважаючи на те, що теорія лінійних дескрипторних систем керування була дуже насиченою за змістом, існує лише кілька вичерпних книг на цю тему, наприклад [3].

Існують деякі інші книги та докторські наукові тези, що стосуються дескрипторних систем, але всі вони зосереджені на дуже особливих темах.

Ми проаналізували різноманітні джерела та з'ясували інформацію щодо досліджень послідовної композиції дескрипторних систем керування.

З цього випливає висновок, що дослідження послідовної композиції дескрипторних систем керування можуть сильно допомогти у розвитку радіотехнічних систем, аналізу їх перехідних режимів, синтезу фільтрів та іншого.

Також було досліджено альтернативний метод каскадної композиції дескрипторних систем [7] та на основі цих досліджень було розроблено та реалізовано алгоритм, який автоматично розраховує коефіцієнти рівнянь (4), (5) скомпонованої дескрипторної системи Φ за допомогою відповідних коефіцієнтів складових систем Φ_1, Φ_2 при послідовній(каскадній) композиції..

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Р. Калман, П. Фалб, М. Арбиб. Очерки по математической теории систем. – М.: Мир, 1971.
2. D.J. Bender, A. Laub. The linear-quadratic optimal regulator for descriptor systems // IEEE Transactions on Automatic Control, vol. AC- 32, no. 8, pp. 672-688, 1987.
3. G.R. Duan. Analysis and Design of Descriptor Linear Systems. – New York, Dordrecht, Heidelberg, London: Springer, 2010.
4. Л.А. Власенко, А.Г. Руткас. Об оптимальном управлении переходными процессами в электрических цепях // Радиоэлектроника и информатика, N1, с. 3-7, 2016.
5. L.A. Vlasenko, A.G. Rutkas, A.A. Chikrii. On a differential game in an abstract parabolic system // Proceedings of the Steklov Institute of Mathematics, vol. 293 (Suppl. 1), pp. 254-269, 2016.
6. L.A. Vlasenko, A.G. Rutkas. Optimal control of undamped Sobolev-type retarded systems // Mathematical Notes, vol. 102, no. 3, pp. 297- 309, 2017.
7. Л.А. Власенко, А.Г. Руткас, В.В. Семенец. Последовательные композиция и декомпозиция дескриптрных систем управления // Проблемы управления и информатики, Т. 50, № 5, с. 5-18, 2018.
8. Ф.Р. Гантмахер. Теория матриц, – М.: Наука, 1966.
9. Л.А. Власенко. Эволюционные модели с неявными и вырожденными дифференциальными уравнениями. – Днепропетровск: Системные технологии, 2006. – 273 с.
10. B.M. Chen, Z. Lin, Y. Shamash. Linear Systems Theory: A Structural Decomposition Approach. – Boston, Basel, Berlin: Birkhäuser, 2004.

11. E.G. Gilbert. Controllability and observability in multivariable control systems: 'Journal of the Society for Industrial and Applied Mathematics. – Control, vol. 1, pp. 128-151, 1963.
12. А.Г. Руткас. Задача Коши для уравнения $Ax'(t) + Bx(t) = f(t)$ // Дифференц. уравнения. – 1975. – 11, N1. – 1996-2010 с.
13. Теорія керування [Електронний ресурс], Вікіпедія. URL: https://uk.wikipedia.org/wiki/Теорія_керування (дата доступу: 21.05.2020).
14. ДСТУ 3008-95. Державний стандарт України. Документація. Звіти у сфері науки і техніки. Структура і правила оформлення [Текст] ДСТУ 3008-95. – Чинний з 01.01.2006. – К.: Держспоживстандарт України, 2005, 38с.