

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЖИТОМИРСЬКИЙ ДЕРЖАВНИЙ ТЕХНОЛОГІЧНИЙ УНІВЕРСИТЕТ**

Ю.О. Подчашинський, Ю.О. Шавурський, О.О. Лугових

**ПРОЕКТУВАННЯ ТА КОНСТРУЮВАННЯ ПРИСТРОЇВ
ТА СИСТЕМ УПРАВЛІННЯ**

Навчальний посібник

Друкується за рішенням
Вченої ради ЖДТУ
(протокол № 9
від 24 вересня 2018р.)

ЖДТУ
2018

УДК 681.5
П44

*Рекомендовано до друку Вченою радою
Житомирського державного технологічного університету
(протокол № 9 від 24 вересня 2018р.)*

Рецензенти: д.т.н., проф. *Грабар І.Г.* (Житомирський національний агроекологічний університет); д.т.н., проф. *Пількевич І.А.* (Житомирський військовий інститут ім. С.П. Корольова); д.ф.-м.н., проф. *Москвін П.П.* (Житомирський державний технологічний університет).

Подчашинський Ю.О., Шавурський Ю.О., Лугових О.О.

П44 Проектування та конструювання пристроїв та систем управління:
Навчальний посібник. – Житомир: ЖДТУ, 2018. – 280 с.

ISBN 978-966-683-506-5

В навчальному посібнику розглянуто загальні принципи проектування та конструювання пристроїв та систем управління, методи формалізації завдання на проектування та методи розробки алгоритмів управління виробничим обладнанням на промислових підприємствах. Проектування пристроїв автоматики та управління пропонується здійснювати у формі скінчених та мікропрограмних автоматів на елементній базі цифрової схемотехніки. Розглянуто питання проектування систем автоматичного регулювання та контролю параметрів об'єктів. Розроблено методи вимірювання та контролю поточних координат, геометричних параметрів та параметрів руху елементів виробничого обладнання та промислових виробів, що виготовляються, на основі комп'ютеризованої обробки їх відеозображень. Для синтезу регуляторів та моделювання роботи систем управління використано пакет прикладних програм MATLAB. Оцінювання поточних координат та параметрів руху об'єктів управління здійснюється на основі фільтра Калмана та експоненціального згладжування результатів вимірювань. Матеріал навчального посібника має практичну направленість та супроводжується прикладами. В посібнику також наведено теоретичні відомості та завдання до лабораторного практикуму з проектування систем управління. Навчальний посібник розраховано на студентів спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології», він також може бути корисним для фахівців, які займаються розробкою систем управління.

Обсяг 280 стор.; 130 іл.; 37 табл.; бібліогр. 50 назв.

УДК 681.5

© Ю.О. Подчашинський, 2018 р.

© Ю.О. Шавурський, 2018 р.

© О.О. Лугових, 2018 р.

ISBN 978-966-683-506-5

ЗМІСТ

	стор.
ВСТУП	6
РОЗДІЛ 1. ЗАГАЛЬНІ ПРИНЦИПИ ПРОЕКТУВАННЯ ПРИСТРОЇВ ТА СИСТЕМ УПРАВЛІННЯ	8
1.1. Системи автоматики та управління, їх класифікація	8
1.2. Основні етапи розвитку технічних систем.....	11
1.3. Задачі проектування.....	12
1.4. Етапи проектування	14
1.5. Об'єкти проектування.....	17
1.6. Класифікація параметрів об'єктів проектування.....	18
1.7. Методи проектування	21
1.8. Спадкоємність при проектуванні	23
1.9. Винахідництво та проектування.....	24
1.10. Прийняття рішення	25
1.11. Оцінка ефективності системи	26
1.12. Технічне завдання	27
1.13. Контрольні питання до розділу 1	29
РОЗДІЛ 2. ФОРМАЛІЗАЦІЯ ЗАВДАННЯ НА ПРОЕКТУВАННЯ ТА АЛГОРИТМУ УПРАВЛІННЯ	30
2.1. Умови застосування систем автоматики та управління.....	30
2.2. Формалізація завдання на проектування	30
2.3. Розробка структурної схеми системи управління	33
2.4. Формалізація алгоритму управління.....	35
2.5. Контрольні питання до розділу 2	43
РОЗДІЛ 3. ПРОЕКТУВАННЯ СКІНЧЕНИХ АВТОМАТІВ	44
3.1. Загальні відомості про скінчені автомати	44
3.2. Проектування автоматів з паралельними входами.....	45
3.2.1. Проектування автоматів на зворотних зв'язках	45
3.2.2. Проектування автоматів на тригерах.....	53
3.3. Проектування автоматів з послідовними входами	60
3.3.1. Принцип проектування автоматів	60
3.3.2. Оптимальне кодування станів автоматів.....	65
3.3.3. Проектування багатовходових автоматів.....	68
3.4. Контрольні питання до розділу 3	74

РОЗДІЛ 4. ПРОЕКТУВАННЯ МІКРОПРОГРАМНИХ

АВТОМАТИВ.....	75
4.1. Загальні відомості про мікропрограмні автомати	75
4.2. Форми представлення алгоритму роботи мікропрограмних автоматів	76
4.3. Структурна схема мікропрограмних автоматів	79
4.4. Складання логічної схеми алгоритму	80
4.5. Спрощення логічної схеми алгоритму.....	83
4.6. Формування мікрокоманд	88
4.7. Часткові логічні схеми алгоритмів та їх об'єднання.....	91
4.8. Практичне проектування мікропрограмних автоматів	93
4.9. Контрольні питання до розділу 4	100

РОЗДІЛ 5. ПРИКЛАДИ РОЗРОБКИ СТРУКТУРНОЇ ТА

ПРИНЦИПОВОЇ СХЕМИ СИСТЕМИ УПРАВЛІННЯ.....	101
5.1. Розробка структурної схеми системи управління	101
5.2. Розробка принципової схеми системи управління	109
5.3. Контрольні питання до розділу 5	119

РОЗДІЛ 6. ПРОЕКТУВАННЯ СИСТЕМ АВТОМАТИЧНОГО РЕГУЛЮВАННЯ ТА КОНТРОЛЮ ПАРАМЕТРІВ

ОБ'ЄКТІВ	120
6.1. Загальна характеристика систем автоматичного регулювання.	120
6.2. Основні терміни та визначення	121
6.3. Вибір регулятора	123
6.4. Вибір датчика	131
6.5. Вибір регулюючого органу	132
6.6. Позиційні регулятори	134
6.7. Вибір регуляторів в програмі MATLAB.....	137
6.8. Синтез регуляторів методом кореневих годографів	139
6.9. Синтез регуляторів частотними методами.....	140
6.10. Синтез регуляторів на основі моделювання об'єктів та систем в просторі стану.....	141
6.11. Методи оцінювання параметрів руху об'єктів вимірювань	142
6.12. Визначення координат і параметрів руху об'єктів в реальному часі.....	146
6.13. Контрольні питання до розділу 6	162

РОЗДІЛ 7. ЛАБОРАТОРНИЙ ПРАКТИКУМ З ПРОЕКТУВАННЯ СИСТЕМ УПРАВЛІННЯ.....	163
7.1. Лабораторна робота 1. Введення в програму MATLAB математичних моделей систем управління та дослідження їх характеристик.....	163
7.2. Лабораторна робота 2. Перетворення математичних моделей систем управління в програмі MATLAB.....	180
7.3. Лабораторна робота 3. Синтез та моделювання роботи регулятора для системи управління.....	190
7.4. Лабораторна робота 4. Синтез регуляторів для системи управління частотними методами.....	201
7.5. Лабораторна робота 5. Методи перетворення неперервної системи автоматичного управління в цифрову систему.....	208
7.6. Лабораторна робота 6. Перетворення математичних моделей цифрової системи управління в програмі MATLAB.....	225
7.7. Лабораторна робота 7. Синтез регуляторів для цифрової системи управління часовими методами.....	230
7.8. Лабораторна робота 8. Синтез регуляторів для цифрової системи управління частотними методами.....	235
7.9. Лабораторна робота 9. Застосування фільтра Калмана для вимірювання поточних координат та параметрів руху об'єктів управління.....	241
7.10. Лабораторна робота 10. Застосування експоненціального згладжування для вимірювання поточних координат об'єктів управління.....	252
ДОДАТКИ.....	267
Додаток А. Структурні схеми систем автоматичного управління...	267
Додаток Б. Параметри структурних схем систем автоматичного управління.....	271
Додаток В. Програма експоненціального згладжування та оцінювання значень поточних координат та параметрів руху об'єктів.....	272
СПИСОК ЛІТЕРАТУРНИХ ДЖЕРЕЛ.....	277

ВСТУП

Навчальна дисципліна «Проектування та конструювання пристроїв та систем управління» є однією з провідних дисциплін спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології», яка великою мірою визначає якість підготовки фахівців з даної спеціальності.

Проектування можна віднести до найпоширеніших видів діяльності людини. Вже створення перших, навіть дуже примітивних, знарядь праці вимагало від людини аналізу зовнішнього середовища, винахідництва та вибору найкращого варіанту, тобто прийняття рішення. З розвитком людства ускладнювалися задачі, розвивалася технічна база, зростав рівень технічних пристроїв та систем.

Науково-технічна революція корінним чином змінила всі технічні системи, в тому числі і пристрої та системи управління. Характерним стало застосування в них цифрових обчислювальних машин, багаторівневих систем, використання людино-машинних комплексів, значне підвищення вимог до надійності та ефективності. Поряд з цим визначилася тенденція до швидкого морального старіння та необхідності постійного оновлення технічних систем. З цих причин необхідно скорочувати час проектування пристроїв та систем управління, в тому числі – на основі розробки і застосування систем автоматизованого проектування.

Сучасний розвиток засобів управління характеризується широким впровадженням мікроелектроніки. Тільки застосування інтегральних мікросхем, мікропроцесорів та мікроконтролерів дозволяє усунути протиріччя між складністю та надійністю пристроїв. Використання інтегральних мікросхем дозволило не тільки зменшити розміри та масу апаратури, збільшити її надійність та строк служби, але й застосовувати нові принципи проектування, які забезпечують можливість максимально автоматизувати процеси проектування та конструювання апаратури.

Наявність готових блоків та вузлів у вигляді інтегральних мікросхем переміщує процес проектування в область верхніх ієрархічних рівнів, де на передньому плані знаходяться загальні проблеми оптимального проектування систем управління в цілому.

В навчальному посібнику розглянуто загальні принципи проектування та конструювання пристроїв та систем управління, методи формалізації завдання на проектування та методи розробки алгоритмів управління виробничим обладнанням на промислових підприємствах. Багато процедур проектування не мають формального опису і вимагають від фахівців вміння логічно мислити. При розробці систем управління завжди є можливість вибору серед декількох варіантів рішення, що вимагає вміння оцінювати ці варіанти з точки зору вартості, енергоспоживання, габаритів, точності та швидкодії.

Проектування пристроїв автоматики та управління пропонується здійснювати у формі скінчених та мікропрограмних автоматів на елементній базі цифрової схемотехніки. Наявність аналого-цифрових перетворювачів (АЦП) дозволяє використовувати елементну базу цифрової схемотехніки і обробляти сигнали від аналогових вимірювальних перетворювачів фізичних величин. При цьому, чим раніше буде зроблено перехід від аналогових сигналів до цифрових, тим кращі значення параметрів системи управління можна забезпечити.

Викладені в навчальному посібнику методи проектування пристроїв та систем управління забезпечують їх функціонування за заданим алгоритмом та мінімальні апаратні витрати. Широкий діапазон логічних та цифрових елементів, від елементів малої інтеграції до великих інтегральних схем, мікропроцесорів та мікроконтролерів, дозволяє розробляти на їх основі пристрої та системи управління практично будь-якої складності.

Також розглянуто питання проектування систем автоматичного регулювання та контролю параметрів об'єктів. Розроблено методи вимірювання та контролю поточних координат, геометричних параметрів та параметрів руху елементів виробничого обладнання та промислових виробів, що виготовляються, на основі комп'ютеризованої обробки їх відеозображень. Для синтезу регуляторів та моделювання роботи систем управління використано пакет прикладних програм MATLAB. Оцінювання поточних координат та параметрів руху об'єктів управління здійснюється на основі фільтра Калмана та експоненціального згладжування результатів вимірювань.

При написанні посібника вважалося, що студенти добре володіють апаратом дискретної математики, комп'ютерної логіки, знають елементну базу цифрової схемотехніки, знайомі з методами розробки алгоритмів та основами програмування, володіють основами роботи з пакетом прикладних програм MATLAB.

Навчальний посібник не охоплює всіх проблем проектування та практичної реалізації систем управління. Але він закладає у студентів знання базових принципів, методів та практичних навичок проектування, що дозволить їм у подальшій професійній діяльності самовдосконалюватися та вирішувати складні завдання з розробки систем управління на сучасній програмній та технічній базі.

Матеріал навчального посібника має практичну направленість та супроводжується прикладами. В посібнику також наведено теоретичні відомості та завдання до лабораторного практикуму з проектування систем управління. Навчальний посібник розраховано на студентів спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології», він також може бути корисним для фахівців, які займаються розробкою систем управління.

Автори висловлюють вдячність рецензентам за підтримку, корисні поради та зауваження, що дозволили покращити якість навчального посібника.

РОЗДІЛ 1

ЗАГАЛЬНІ ПРИНЦИПИ ПРОЕКТУВАННЯ ПРИСТРОЇВ ТА СИСТЕМ УПРАВЛІННЯ

1.1. Системи автоматики та управління, їх класифікація

Під системою прийнято розуміти впорядкований пристрій, який складається з взаємопов'язаних частин, що діють як одне ціле, і призначений для досягнення певної цілі. Це визначення не є повним та точним [1].

У загальному випадку системою може бути будь-який фізичний об'єкт, який складається з певної кількості взаємопов'язаних елементів. Якщо стан системи змінюється в часі, то така система називається динамічною.

Системи можна класифікувати за різними ознаками, серед яких можна виділити два основних: складність системи та характер їх функціонування.

Простими вважаються системи, які не мають розгалуженої структури з кількістю взаємопов'язаних елементів від 10 до 1000.

До складних відносять системи з розвинутою ієрархічною структурою і кількістю елементів від 10^4 до 10^7 .

Подібне визначення є досить умовним. Часто до складних систем відносять такі, які неможливо коректно описати математично, або системи, для вивчення яких необхідно вирішувати задачі з великим об'ємом обчислень.

Дуже складні системи називають великими системами.

З функціональної точки зору системи можуть бути детермінованими, в яких елементи взаємодіють точно визначеним чином, та ймовірнісними (стохастичними), в яких передбачити майбутню поведінку системи можливо лише з певним ступенем вірогідності.

Системи управління (СУ) можна поділити на системи автоматичного управління (САУ) та на автоматизовані системи управління (АСУ). В САУ всі функції виконуються без участі оператора, тоді як в АСУ вони виконуються спільними зусиллями як системи, так і оператора [1 – 3].

На перший погляд здається, що САУ більш складна система, ніж АСУ. Безумовно, зустрічаються САУ і АСУ різної складності, але в цілому більш складною треба вважати АСУ. Присутність оператора пояснюється тим, що в складних об'єктах не всі функції управління можуть бути передані системі, особливо в умовах неповного знання законів функціонування об'єкта. Оператор, з його вмінням, досвідом та інтуїцією, може прийняти більш правильне рішення в різних позаштатних ситуаціях. Крім того, введення додаткових органів управління не завжди виправдано з економічної точки зору.

СУ можуть бути однорівневі та багаторівневі. Однорівневі системи можуть бути централізовані та децентралізовані.

Для правильного вибору структури СУ необхідно чітко уявляти позитивні та негативні сторони цих систем.

Припускаємо, що СУ повинна забезпечити управління n - об'єктами (n - частинами об'єкта). Це означає, що в загальній задачі управління можна виділити n задач.

В централізованій структурі, яка зображена на рис. 1.1, а, СУ одна і необхідна висока швидкість обробки інформації для забезпечення вирішення всіх задач. В децентралізованій структурі (рис. 1.1, б) всі системи управління працюють паралельно і продуктивність кожної з них може бути меншою.

Вартість централізованої структури нижча, тому що одне і те ж обладнання використовується для вирішення всіх задач.

По надійності надати перевагу неможливо жодному з наведених типів структури. В централізованій структурі кількість елементів менша і їх надійність вища. Але відмова одного з елементів у ній призводить до зупинки всієї системи. В децентралізованій структурі елементів більше, але відмова одного з елементів у ній виключає можливість управління лише одного об'єкта.

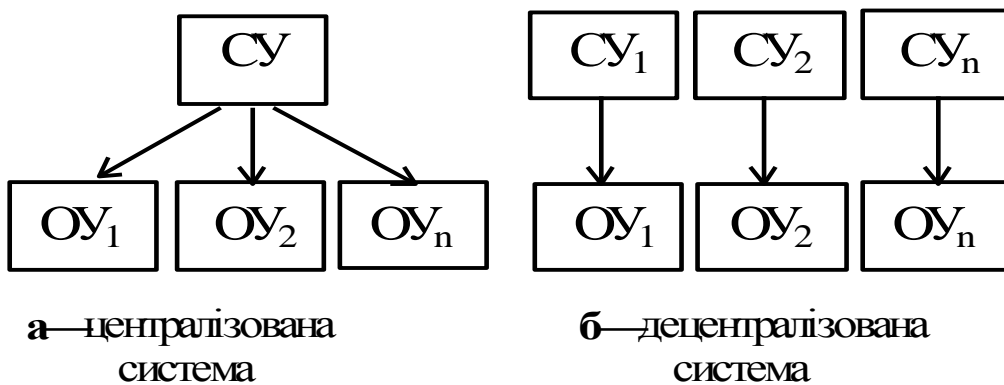


Рис. 1.1

З точки зору ремонтпридатності перевагу слід надати децентралізованій структурі, яка більш проста і до того ж розчленована.

Обидві структури слід розглядати як граничні з можливих модифікацій.

Дуже часто більш вигідними є багаторівневі структури, які поєднують в собі позитивні сторони обох однорівневих структур. Схематично багаторівнева структура зображена на рис. 1.2. Безпосередньо біля об'єкта розміщується термінальні системи управління ТСУ1 ÷ ТСУ_n. Вони утворюють нижній рівень управління. Найбільш складні задачі оптимального управління вирішує центральна система управління ЦСУ, яка утворює верхній рівень управління. Як раз тут застосовується найскладніше обладнання і його централізація дає найбільший ефект. Роль ЦСУ, як правило, виконує ЕОМ. ТСУ також можна виконати на базі мікро- або міні-ЕОМ, або на базі програмованих контролерів.

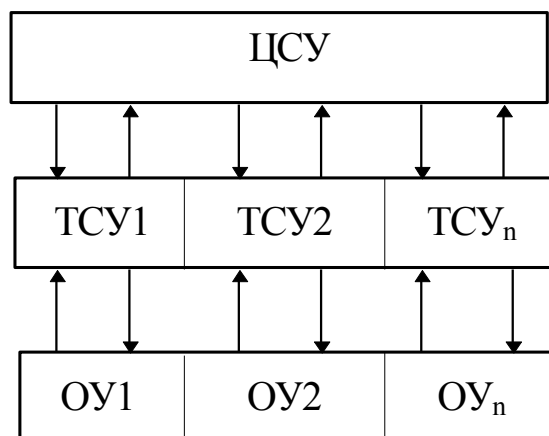


Рис. 1.2

Основний обмін інформацією здійснюється між об'єктами та ТСУ. В ЦСУ передається тільки найбільш важлива інформація, що розвантажує лінії зв'язку між ЦСУ та ТСУ.

ТСУ можуть мати запам'ятовуючі пристрої, що робить їх автономними. Через це, в разі пошкодження ЦСУ, вся система розпадається на n незалежних систем, що не заважає проведенню процесу управління. В разі пошкодження ТСУ, її роль може взяти на себе ЦСУ.

Додаткові переваги багаторівневої структури в тому, що її впровадження в експлуатацію можна здійснювати почергово. Доки впроваджується ЦСУ, ТСУ працюють і приносять економічний ефект.

Незалежно від типу, системи управління можна розглядати як сукупність блоків (пристроїв) з різним функціональним призначенням. Можна виділити наступні основні блоки САУ або АСУ. Блок отримання контрольної інформації. Цю групу складають пристрої, які заміряють значення параметру і які називаються датчиками або первісними перетворювачами. Блок передачі контрольної інформації. Сюди відносять перетворювачі виду та рівня сигналів, модулятори, демодулятори, підсилювачі, шифратори, дешифратори, лінії зв'язку та інші. Блок обробки інформації та видачі управляючих сигналів. Це найбільш складна частина СУ, яка може складатися з пристроїв пам'яті, арифметично-логічних пристроїв, мікропроцесорів, ЕОМ і т. п. Якраз в цій групі застосування програмних засобів управління є найбільш перспективним, аж до застосування управляючих ЕОМ. Блок передачі командної інформації. Ця група має багато спільного з другою групою, тільки тут більш актуальні енергетичні проблеми. Блок виконання командної інформації. До цієї групи відносяться різні двигуни, електромагнітні реле, сигнальні лампочки тощо.

Поділ СУ на елементи, пристрої та системи досить умовний. Бурхливий розвиток інтегральної технології весь час змінює межу наведеної класифікації.

1.2. Основні етапи розвитку технічних систем

Будь-яка технічна система виникає не відразу і проходить етапи розвитку, основні з яких показані на рис. 1.3.

Задумка нової системи ніколи не виникає в закінченому вигляді. В процесі подальшої розробки вона може суттєво змінитися.

В процесі проектування нова система повинна бути відображена на папері, тобто необхідно розробити та випустити технічну документацію, по якій розроблена система може бути виготовлена в промислових умовах.

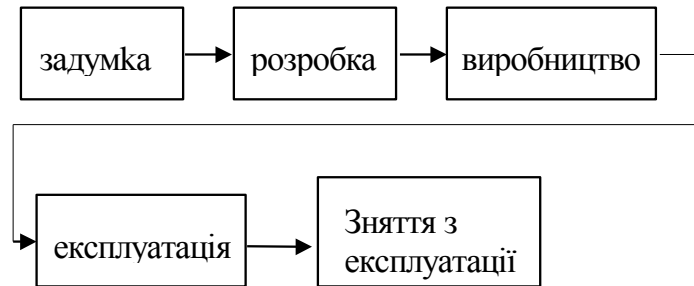


Рис. 1.3

Потім необхідно виготовити дослідні зразки, які проходять необхідну перевірку. Тому етап проектування включає не тільки виготовлення технічної документації, але теоретичну та експериментальну перевірку зразків. Для цього на етапі проектування існує процес виготовлення одиничних та дослідних зразків, що забезпечує проведення необхідних експериментальних робіт, по яким проводять доробку технічної документації та технології виготовлення систем.

Дослідження на етапі проектування охоплює як моделювання та лабораторні дослідження, так і натурні дослідження.

Серійне виробництво системи здійснюється по технічній документації, яка відпрацьована на етапі проектування.

Через те, що технологія серійного виробництва відмінна від технології дослідного виробництва, в процесі серійного виробництва здійснюється доведення документації за участю розробників системи. Основна задача розробників на цьому етапі зводиться до розробки методів оптимізації виробництва та підвищенні його ефективності.

Експлуатація системи з точки зору споживача є основним етапом життя системи. Зусилля розробників повинні бути направлені на те, щоб система на протязі заданого проміжку часу виконувала задані функції. Для цього на етапі проектування розробляються методи та засоби обслуговування системи. Ці засоби включають систему контролю та відновлення працездатності системи. Тому зняття системи з експлуатації пов'язано з її моральним старінням, коли подальша експлуатація системи стає неефективною.

1.3. Задачі проектування

Проектування є високо інтелектуальною роботою, творчістю, яке вимагає застосування різноманітних знань.

Задачею проектування є розробка, при деяких обмеженнях, обумовлених способом рішення, систем, які забезпечують оптимальне виконання поставленої задачі, при деяких обмеженнях, що накладаються на рішення [1].

В цьому визначенні двічі зустрічаються слова “ при деяких обмеженнях”. Перше обмеження стосується методу проектування, друге — системи, що проектується.

Основними обмеженнями при проектуванні є: наявність знань та навиків, в тому числі технологічних; наявність необхідних матеріалів та комплектуючих елементів; можливості обладнання; наявність обчислювальної техніки та терміни проектування.

При сучасних темпах науково-технічного прогресу скорочення термінів проектування стає однією з найголовніших вимог до процесу проектування. При великих термінах проектування система може морально застаріти ще до її виготовлення.

Таким чином, обмеження на метод проектування визначають можливість розробки в задані терміни. Обмеження на систему визначаються можливістю її виготовлення.

Задачі або цілі проектування, як очікуваний результат майбутньої роботи, є ініціатором всієї іншої роботи. Як правило, вона має розгорнутий характер від формулювання загального плану до більш чи менш конкретних задач, які виникають в процесі їх багаторазових уточнень методами послідовних наближень [3 – 4].

Цілі характеризуються своїми загальними якісними описами (критеріями) найбільш суттєвих сторін цих якостей (точність, надійність, економічність і т. п.).

Первісне формулювання цілі передбачає тільки загальний напрямок майбутньої роботи. Однозначні результати, шляхи та засоби їх досягнення поки що не передбачаються. Навпаки, допускається багатоваріантний розвиток подій в заданому напрямку. Він і не може бути іншим в силу неоднозначності, що об’єктивно впливає на початковому етапі проектування.

Але результат роботи повинен бути однозначним. Тому, після формулювання загальної цілі, здійснюється перехід до побудови дерева (ієрархічного графа) цілей, коли загальна ціль розподіляється на логічно взаємопов’язані цілі, які забезпечують виконання поставленої задачі. Останні поступово, по ступенях ієрархії, втрачають загальність і нарощують конкретність. Цей процес декомпозиції загальної цілі продовжується, доки не з’являться конкретні шляхи їх вирішення, а через них – шляхи та засоби вирішення загальної цілі, - рішення проблеми в цілому.

Діалектика взаємовідносин цілей та засобів така, що цілі завжди мають в собі певні вказівки про засоби. Загальна ціль не досить конкретна і має мало вказівок на те, як вирішити задачу. Дерево цілей призводить на нижніх рівнях до досить конкретних цілей і допускає не тільки якісний, але й кількісний опис. Таким чином, дерево цілей стає не тільки інструментом більш точного опису плануючих результатів, а й вихідною базою для формування складу об'єкта.

Такий хід проектування називається блочно-ієрархічним і є основним при проектуванні. Недоліки такого підходу полягають в тому, що на кожному рівні робота ведеться з не до кінця визначеним об'єктом. Дійсно, як елементи на i -му рівні використовуються досить складні об'єкти, які на $(i+1)$ -му рівні розглядаються як системи. На i -му рівні ці елементи ще невідомі, тому що структура цього рівня формується до того, як будуть розроблені його елементи. Отже, рішення приймається при неповній інформації, тобто без чіткого обґрунтування. Тому оптимальність можна досягнути тільки на окремих рівнях при обмеженнях, які не є принципово необхідними. І все ж в цілому більш вдалої альтернативи блочно-ієрархічному підходу немає і ми змушені погоджуватися на можливі відхилення від оптимальних рішень.

В літературі [7] згадується про метод проектування, в якому систему формують, починаючи з рівня відомих елементів. Такий підхід може дати позитивний результат тільки для нескладних пристроїв, тому що сформулювати алгоритм складної системи в рівнях елементів практично неможливо.

Найважливішою задачею проектування є розробка повного комплексу технічної документації на систему. Ця документація повинна забезпечити можливість промислового виготовлення системи та забезпечити надійну її експлуатацію в заданих умовах.

В результаті проектування випускається великий об'єм технічної документації, склад якої визначається єдиною системою конструкторської документації (ЄСКД). ЄСКД – це сукупність загальних для всіх галузей промисловості взаємопов'язаних правил виконання, оформлення та використання документації на всіх етапах розробки, виготовлення, експлуатації та ремонту промислових виробів.

ЄСКД можна умовно розподілити на три групи:

- 1) стандарти на правила розробки та класифікації конструкторських документів;
- 2) стандарти на правила виконання та оформлення конструкторських документів;
- 3) стандарти на правила відношення та використання конструкторської документації.

Технічна документація, яка випускається в процесі проектування, розподіляється на наступні категорії:

- 1) схемна;
- 2) конструкторська;
- 3) монтажна;
- 4) текстова;
- 5) технологічна;
- 6) експлуатаційна.

Перші чотири категорії документації є відображенням ідеї та принципу, що закладені в систему при її проектуванні, і відповідають на питання, що повинно бути виготовлено.

Схеми розподіляються на принципові, функціональні та структурні.

Принципові схеми визначають повний набір базових елементів та зв'язків між ними і дають детальне уявлення принципу дії системи.

Функціональні схеми пояснюють хід певних процесів в системі або її частинах і враховують найбільш суттєві фактори та функціональні частини.

Структурні схеми дають найбільш загальне і найменш детальне уявлення про об'єкт, його основні функціональні частини, їх призначення і взаємозв'язки.

Блочно-ієрархічне уявлення про об'єкт проектування можна назвати поділом його на горизонтальні рівні. В свою чергу на горизонтальних рівнях можна виділити задачі проектування схем, конструкцій та технологій. Їх називають ще вертикальними рівнями.

Технологічна документація дає уявлення про методи виготовлення системи.

Експлуатаційна документація включає основні документи схемної, конструкторської та текстової документації і повинна забезпечити грамотну експлуатацію системи.

1.4. Етапи проектування

Процес проектування укрупнено можна зобразити рис. 1.4. Етапи проектування пов'язані двосторонніми стрілками. Це вказує на те, що етапи можуть повторюватися, тобто процес проектування є ітераційним процесом.

В процесі проектування неодноразово приймаються рішення. Прийняття рішення є одним із визначальних моментів проектування систем.

Особливо суттєве прийняття рішення на етапах 1-3, тому що вони в значній мірі визначають якість проекту і системи. Наведені на рис. 1.4 етапи групують в декілька стадій проектування, а саме:

- 1) розробка технічного завдання на проектування;
- 2) попереднє проектування (технічна пропозиція);
- 3) ескізне проектування;
- 4) технічне (робоче) проектування;
- 5) серійне виготовлення;
- 6) експлуатація.

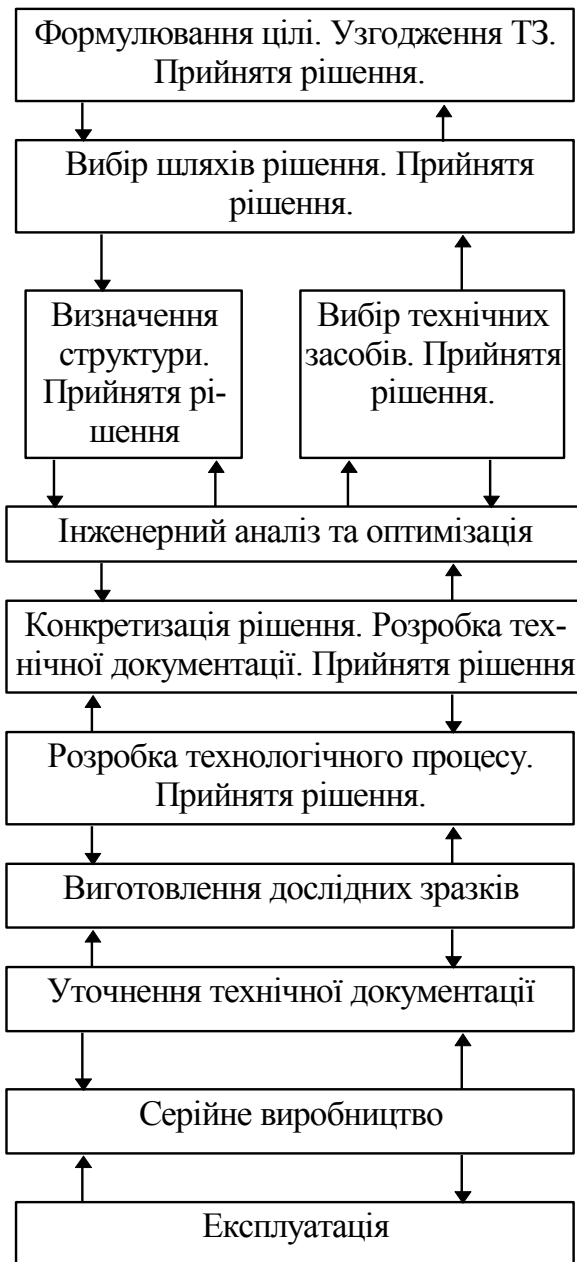


Рис. 1.4

Розробка технічного завдання є початковим і відповідальним етапом проектування. Воно багато в чому визначає як рівень так і реалізацію системи, яка розробляється.

Технічна пропозиція включає рішення наступних питань:

1) виявлення варіантів можливих рішень, встановлення особливостей цих варіантів (принципи дії, розміщення функціональних складових частин тощо), їх конструктивну доробку. Глибина такої доробки повинна бути достатньою для порівняльної оцінки варіантів;

2) перевірка варіантів на патентну чистоту та конкурентоспроможність, оформлення заявок на винаходи;

3) перевірка відповідності варіантів вимогам техніки безпеки та виробничої санітарії;

4) порівняльна оцінка варіантів. Порівняння проводиться по показникам якості виробу, наприклад, надійності, економічним, ергономічним або по технологічним (питома трудомісткість, матеріалоемність тощо).

Таке проектування відносять до стадії науково-дослідних робіт (НДР).

Ескізне проектування включає вирішення наступних питань:

- 1) виконання варіантів можливих рішень, встановлення особливостей варіантів (характеристики варіантів складових частин тощо);
- 2) попереднє вирішення питань упаковки та транспортування виробу;
- 3) виготовлення та дослідження макетів для перевірки принципів дії виробу або його складових частин;
- 4) розробку та обґрунтування технічних рішень для забезпечення показників надійності;
- 5) оцінка виробу на технологічність та вірність вибору засобів та методів контролю;
- 6) оцінка виробу по показникам стандартизації та уніфікації;
- 7) оцінку виробу на відповідність вимогам ергономіки та технічної естетики;
- 8) перевірка виробу на патентну чистоту та оформлення заявок на винаходи;
- 9) перевірка відповідності виробу вимогам техніки безпеки та промислової санітарії;
- 10) порівняльну оцінку варіантів по питанням метрологічного забезпечення (можливості вибору методів та засобів вимірювання).

Результатом ескізного проекту є детальна доробка можливості розробки необхідної системи.

Ескізне проектування та наступні етапи відносять до стадії дослідно-конструкторських робіт (ДКР). На етапі технічного проектування проводиться детальна розробка схемних, конструкторських та технологічних рішень. В процесі серійного виробництва здійснюється підсумкова доводка прийнятих технічних рішень і відпрацювання технології з урахуванням особливостей виробництва. В процесі експлуатації розробник отримує інформацію, яка дозволяє вносити необхідні корективи з метою доведення параметрів системи до заданих.

Згідно діючим стандартам, документація, яка відноситься до технічної пропозиції, має літеру "П"; документація, яка відноситься до ескізного проекту, має літеру "Е"; документація, яка відноситься до технічного проекту, має літеру "Т". Робоча документація має наступні літери:

- дослідного зразку — "О". Наступні зразки — "О1", "О2" і т.д.;
- установочної серії — "А";
- масового виробництва — "Б".

1.5. Об'єкти проектування

Об'єкти проектування – це конструкції, процеси або системи, що є майбутніми засобами досягнення цілей. В ході проектування вони існують лише в нашій уяві, в попередніх описах або в моделях. При цьому наше уявлення про них послідовно уточнюється в процесі проектування від первісного якісного до підсумкового документу – проекту [1 – 6].

Якщо, наприклад, ціль – вирішення житлової проблеми для ста сімей, то засобом її вирішення можна вважати будівництво будинку на сто квартир. Останній стає таким чином об'єктом проектування. Таке визначення об'єкта досить лише для початку роботи. Далі необхідно мати більш точний та повний його опис: кількість поверхів, план квартир, тепло - та водопостачання тощо. На ці та інші запитання буде отримано відповідь на наступних етапах проектування. Повне уявлення про нього дає завершений проект.

Як бачимо, зміст поняття “об'єкт проектування” суттєво пов'язаний з цілями та процесом проектування. Він розвивається по етапах цього процесу і повинен послідовно відповідати цільовим установкам ієрархії цілей. Тим самим здійснюється єдність цілі, об'єкта та процесу проектування, взятих в їх розвитку від первісної постановки задачі до завершення проектних робіт. Процес проектування в цій тріаді може розглядатися як алгоритм послідовного уточнення обґрунтування та вигляду проекту під задану ієрархію задач, що впливають з цільових завдань.

Для реалізації такого цілеспрямованого процесу навколо уявного “об'єкта проектування” необхідні досить глибокі ступені його формалізації та моделювання. Без цього неможливо більш чи менш точний опис об'єкта проектування.

Формалізація опису та моделювання об'єкта проектування досягається в основному за рахунок його структуризації та математичного опису елементів структури. На розвинутих етапах проектування до цих моделей можуть додаватися макети, імітатори, дослідні зразки, які дають експериментальні дані про властивості об'єкта.

Розрізняють три основних види формального опису об'єктів проектування: функціональний, морфологічний та інформаційний.

Функціональний опис дає характеристику призначення об'єкта через його експлуатаційні функції: принцип дії, властивості та особливості, які забезпечують виконання цільових завдань. Функціональні характеристики повинні відповідати цілям проектування, тому що вони формулюються відповідно до логіки ієрархії цілей, тобто від досить загальних функцій до більш детальних часткових.

За функціональним описом виникає потреба в структуризації об'єкта – розподіл його на такі частини, які призначені для виконання функцій, що забезпечують виконання поставлених задач. В результаті цього об'єкт проектування стає складною системою, тобто цілісною єдністю

взаємопов'язаних частин – підсистем, агрегатів та вузлів, конструктивних елементів. Кожна частина системи має своє власне цільове та функціональне призначення, принцип дії, конструктивне виконання і разом з тим, через узгоджену систему цілей, бере участь у створенні єдиного цілого – об'єкта проектування.

Опис частин об'єкта, його структур, геометрії тощо називають морфологічним описом. У загальному вигляді такий опис зводиться до функції типу:

$$M = \{\Sigma, R, C\}, \quad (1.1)$$

де Σ – множина елементів;

R – множина зв'язків;

C – множина структур.

Узагальнена математична модель функціонального опису довільного елемента структури (і структури в цілому) можна представити у вигляді співвідношення:

$$Y = F(s, X, U, V, t), \quad (1.2)$$

де Y – вихідна характеристика (характеристики) елемента (системи);

X – режимні параметри;

U – управляючі параметри;

V – дії навколишнього середовища.

Всі ці параметри залежать від часу t та просторових координат s .

Аналіз подібної моделі необхідно проводити з урахуванням обмежень:

$Y \subset \omega_y$; $X \subset \omega_x$; $U \subset \omega_u$;

$t \subset \omega_t$; $S \subset \omega_s$.

де ω_i – межа допустимих значень відповідних параметрів.

Якість функціонування об'єкта або його частин оцінюють критеріями відповідних цілей. Для кількісної оцінки цих якостей вводять функціональні якості, які дозволяють обчислити швидкодію, надійність, економічність тощо.

У загальному випадку моделі функціонування допускають декілька способів досягнення цілі, що дає можливість постановки задачі оптимізації.

Задачі функціонального та морфологічного опису об'єктів проектування пов'язані з використанням та обробкою значних потоків інформації. Тому необхідний і інформаційний опис об'єктів проектування.

Під інформаційним описом розуміють усі можливі відомості про об'єкт: повідомлення, документи та сигнали, що підлягають обробці, зберіганню та передачі.

Інформаційний опис за структурою відповідає ієрархії формулювання цілей, функцій та структур об'єкта проектування.

1.6. Класифікація параметрів об'єктів проектування

Параметр – величина, яка характеризує властивості або режим роботи об'єкта. Серед параметрів об'єктів проектування слід виділити показники

ефективності, які є кількісною оцінкою ступеня відповідності об'єкта його цільовому призначенню. Показники ефективності розподіляють на показники [1]:

- призначення;
- надійності;
- вартості;
- маси;
- габаритів;
- точності.

В залежності від конкретних умов та типів схем можуть бути і інші показники, з яких ті чи інші відіграють головну роль.

Вихідні параметри – це показники якості, за якими судять про правильне функціонування системи. Тому це поняття аналогічне поняттю “показник ефективності”, але їх застосовують до систем на будь-якому ієрархічному рівні.

Вихідні параметри залежать як від властивостей елементів, так і від зв'язків між ними. Кожний новий тип зв'язку задає нову структуру і призводить до якісних змін в роботі системи. До таких же наслідків призводить і заміна одного типу елементів на інший.

Найчастіше множина варіантів є скінченою множиною. Перехід від одного варіанту до іншого або дискретно змінює значення вихідних параметрів, або змінює сам набір цих параметрів. Якщо структура системи визначена, то її вихідні параметри залежать тільки від параметрів елементів та параметрів зовнішніх умов. Внутрішні параметри – це параметри елементів системи. Зовнішні параметри – це параметри зовнішнього по відношенню до об'єкта середовища, яке так чи інакше впливає на його функціонування. При переході від одного рівня ієрархії до іншого вихідні параметри можуть ставати внутрішніми і навпаки.

Якщо на деякому рівні вплив внутрішніх параметрів не враховується, то варіанти об'єкта є варіантами тільки його структури. Такі об'єкти називаються дискретними.

Всі параметри пов'язані між собою співвідношенням:

$$Y = F(X, Q) \quad (1.3)$$

Існування залежності (1.3) не означає, що вона відома. Найчастіше вона задається в алгебраїчній формі, через вирішення системи рівнянь.

На кожному рівні проектування початковими даними є дані технічного завдання, а результати – технічна документація на блоки даного рівня та технічне завдання на блоки наступного рівня. Проектування зводиться до рішення групи задач, які відносяться до задач синтезу або до задач аналізу.

Синтез технічних об'єктів – це створення нових варіантів структури, а аналіз використовується для оцінки цих варіантів. Для дискретних об'єктів задача синтезу зводиться до задачі визначення їх структури. Для безперервних об'єктів рішення задачі синтезу повинно привести до

визначення структури і внутрішніх параметрів. Якщо серед варіантів структури шукають не будь-який варіант, а найкращий в деякому розумінні, то таку задачу називають структурною оптимізацією. Розрахунки внутрішніх параметрів, оптимальних з позиції деякого критерію, називають параметричною оптимізацією. Найчастіше визначають оптимальні значення тільки деяких параметрів, які називають параметрами, якими управляють.

Задача аналізу зводиться до отримання деякої інформації про властивості об'єкта. За частотою застосування задачі аналізу розподіляються на типові та нетипові. Нетипові задачі аналізу використовуються для отримання деякої додаткової інформації, причому алгоритми рішення цих задач можуть бути різноманітними і наперед невідомими.

Другий принцип класифікації задач аналізу призводить до розподілу їх на групи одноваріантного та багатоваріантного аналізу.

Типовими задачами одноваріантного аналізу є задачі:

- аналіз статичного стану;
- аналіз перехідних процесів;
- аналіз стійкості;
- аналіз стаціонарних режимів коливань.

До типових задач багатоваріантного аналізу відносять задачі:

- статистичний аналіз;
- аналіз чутливості;
- параметрична оптимізація.

На кожному ієрархічному рівні процес проектування уявляється як рішення сукупності задач. Розробка блоку по технічному завданню (ТЗ) починається з синтезу або вибору його структури. Для кожного варіанту структури складають модель об'єкта проектування. Ця модель може бути математичною при машинному проектуванні або фізичною при експериментальній розробці. Числові значення параметрів елементів моделі встановлюють або на основі ручних розрахунків, або беруться орієнтовно на основі досвіду та інтуїції інженера. Далі проводять аналіз моделі, яким перевіряють відповідність вихідних параметрів вимогам ТЗ. Якщо ці параметри не відповідають поставленим вимогам, то проводять параметричну оптимізацію. Якщо це не дає позитивного результату – змінюють структуру об'єкта і знову проводять її аналіз. Таким чином, процес проектування носить ітераційний характер. Тому процедури проектування можуть виконуватися багато разів. Як правило, на перших ітераціях аналіз проводять менш ретельно. На останніх стадіях стає виправданим трудомісткий статистичний аналіз.

В конкретних ситуаціях можуть бути відхилення від вказаної послідовності. В деяких випадках структура блоку може бути заданою або відомою. Тоді процедура синтезу виключається. Задача параметричної оптимізації може бути замінена простою задачею розрахунку внутрішніх

параметрів, якщо для досягнення екстремуму маємо допустимий ступінь умов працездатності.

1.7. Методи проектування

При проектуванні необхідно вибрати методи та засоби рішення проектних задач, які забезпечують найкраще досягнення цілей. Останні можуть розумітися як мінімальні строки проектування, як мінімум матеріальних витрат, як оптимальність проектних рішень.

У своїй діяльності інженери використовують в розумному поєднанні експериментальні, розрахункові та інтуїтивно-евристичні методи проектування. Інтуїція та досвід необхідні насамперед при вирішенні задач синтезу структури більшості технічних об'єктів. Для задач визначення значень внутрішніх параметрів використовують розрахункові та експериментальні методи.

Розрахункові ручні методи проектування можуть дати лише приблизні значення параметрів, які слід розглядати як початкові і які вимагають корекції за допомогою інших методів. Це пов'язано з малою точністю ручних методів розрахунків. Дійсно, процеси в досить складних технічних об'єктах описуються системами рівнянь високого порядку, причому в загальному випадку ці рівняння нелінійні.

Перехідні процеси в електронних схемах описуються системами звичайних диференціальних рівнянь, порядок яких дорівнює кількості реактивних елементів. Для схеми середньої інтеграції порядок рівнянь складає від 50 до 500. Аналітичне вирішення системи звичайних диференціальних рівнянь вдається отримати практично лише тоді, коли їх порядок не більше двох, а рівняння лінійні. Це показує, що ручні методи застосовують лише при значних спрощеннях, що й обумовлює приблизний характер отриманих рішень. Тому аналіз працездатності неможливо проводити ручним методом. Через це неминуче було застосування експериментальних методів, тобто методу макетування. На фізичній моделі визначалось виконання умов працездатності, проводилася зміна внутрішніх параметрів, часткова зміна структури. Звичайно, на макеті допустимі не всі зміни, тому що ціна деяких змін надто висока. У багатьох випадках експериментування на досить високих рівнях неможливе, тому що вартість виготовлення макету дорівнює вартості дослідного зразка і погодитися на багаторазове його виготовлення не завжди можливо. Через це при традиційному підході задачі оптимізації практично не вирішувалися – задовольнялися отриманням першого дієздатного варіанту.

Машинні методи з'явилися насамперед як результат устремління замінити дороге та довготривале фізичне моделювання математичним моделюванням. В цих методах, замість макету, використовують математичну модель технічного об'єкта.

Математична модель – це сукупність математичних об’єктів (чисел, змінних, векторів, множин і т. п.) та співвідношень між ними, які адекватно відображають деякі властивості об’єкта, що проектується. В процесі проектування застосовують ті математичні моделі, які відображають найбільш суттєві з точки зору інженера якості об’єкта.

Відсутність вимог суттєвих спрощень може дати необхідну точність моделі. Наявність ЕОМ з відповідною швидкістю та достатнім об’ємом оперативної пам’яті дає можливість проаналізувати модель за відносно малий проміжок часу. В той же час для математичної моделі характерна легкість зміни будь-яких параметрів, що дозволяє виконати більш повний аналіз.

Таким чином, в схемі процесу проектування з появою машинних методів змінився зміст більшості процедур. Насамперед, в процедурі складання моделі фізичне моделювання змінилося на математичне, а процедура “аналіз” перетворилася на вирішення системи рівнянь.

Наступний крок було зроблено в напрямку алгоритмізації процедури “зміна управляючих параметрів”. Їх зміну вдалося зробити ціле направленою. Сучасні САПР можуть самі створювати математичні моделі за деяким описом об’єкта.

Іншою важливою задачею, що вирішується на ЕОМ, є задача оформлення технічної документації. Це пов’язано з розвитком машинної графіки.

Математичні моделі, в яких фігурують тільки вхідні, внутрішні та зовнішні параметри, (аналітичні моделі) зустрічаються досить рідко. В загальному випадку рівняння математичної моделі пов’язують фізичні величини, які характеризують потенціальну та кінетичну енергію і які не відносяться до перерахованих вище параметрів. Ці величини називаються фазовими змінними (наприклад, напруга та струм в електричних системах), а їх вектор задає точку в просторі, який називається фазовим простором.

При складанні математичної моделі в рівняннях можуть бути не всі фазові змінні, а тільки частина їх. Такі змінні називаються базовими координатами. Через них може бути знайдена решта змінних.

В загальному випадку математична модель може мати вигляд:

$$\psi = F(dV/dt, V, t) = 0 \quad (1.4)$$

Параметри елементів теж можуть входити в математичну модель, але у вигляді коефіцієнтів при змінних. Не всі вихідні параметри відносяться до групи функціоналів. Частина важливих якостей об’єкта характеризують вихідні параметри, які називають граничними (наприклад, максимальне навантаження). Під граничними вихідними параметрами розуміють граничні значення зовнішніх параметрів, при яких ще виконується та чи інша ознака правильності функціонування об’єкта.

Таким чином, при машинному аналізі безперервних об’єктів розрізняють три наступні стадії:

- формування математичної моделі об'єкта;
- вирішення системи рівнянь моделі;
- обчислення вихідних параметрів.

Першу стадію називають моделюванням об'єкта, а дві наступні – власне аналізом.

1.8. Спадкоємність при проектуванні

Розробник завжди прагне отримати характеристики системи, які були б кращі за характеристики старої системи. Але це прагнення стримується обмеженими термінами розробки та наявними можливостями.

Виходом з цього протиріччя є раціональна спадкоємність системи, яка розробляється, і старої системи. Ця спадкоємність може охоплювати як досить великі частини, так і відносно дрібні, наприклад, блоки, вузли та елементи.

Основний ефект при використанні спадкоємності досягається за рахунок зменшенню часу на розробку та промислове виготовлення великої кількості конструктивних елементів. Повне оновлення конструкції і матеріальної бази технологічних процесів виправдано лише тоді, коли застаріла конструкція принципово не може бути використана в новій системі.

Однією з основних складових спадкоємності є уніфікація, нормалізація та стандартизація.

Уніфікація є першим ступенем спадкоємності і являє собою процес зменшення різноманітності конструкцій, призначених для виконання одних і тих же або близьких функцій (наприклад, кількість типорозмірів корпусів приладів).

Нормалізація є більш високою ступеню спадкоємності. Основні вимоги нормалізації зводяться до застосування вже освоєних промисловістю приладів, блоків та деталей, а також обмежену номенклатуру елементів та готових виробів.

Нормалізація, як правило, стосується матеріалів, деталей, вузлів та типових виробів. Вимоги нормалізації найбільш широко застосовують до деталей з'єднання, до матеріалів, до монтажних проводів, до монтажних деталей. В меншій мірі вдається нормалізувати складні функціональні блоки, особливо коли вони виконують складні функції.

Стандартизація є метод обмеження різноманітності якісних показників продукції, класифікації, кодування, термінології, технічних вимог, методів випробувань, вимог до упаковки, транспортування тощо.

Використання стандартів при розробці та виготовленню систем є обов'язковим і це значною мірою полегшує процес проектування.

Крім державних, існують міжнародні стандарти, використання яких є обов'язковим, якщо продукція буде продаватися за кордоном.

Оцінка доцільності рівня спадкоємності при проектуванні проводиться методом дослідження операцій, тому що ця задача близька до задачі заміни

технічних пристроїв. Задача заміни технічних пристроїв зводиться до визначення термінів та порядку заміни технічних пристроїв з урахуванням їх фізичного та морального зносу. Методи рішення задач заміни обладнання використовують для рішення задач визначення оптимальних термінів розробки нових видів технічних пристроїв.

1.9. Винахідництво та проектування

Устремління покращити параметри системи призводить до пошуків нових принципів та технічних засобів реалізації, що складає задачу винахідництва. Необхідність в пошуку нових принципів та пристроїв тим більша, чим вищі вимоги до системи, що розробляється. Винахідництво найбільше проявляється на початкових етапах проектування, коли створюються основи майбутньої системи.

Під винахідництвом розуміють вміння знаходити нові ідеї та принципи рішення інженерних задач. Це вміння лежить в області евристики і потребує широких знань, досвіду, вміння зв'язувати між собою різноманітні факти та явища.

Винахід характеризується двома основними ознаками — новизна та корисність. Першій ознаці відповідає нове, яке має суттєву відміну технічного рішення, другій — позитивний ефект застосування.

В практиці проектування об'єктом винаходу може бути новий спосіб або пристрій, а також використання відомих способів або пристроїв по новому призначенню.

Винахідництво складається з наступних основних етапів:

- 1) підготовка — накопичення знань та формулювання задачі;
- 2) концентрація зусилля — клопітка робота з метою пошуку рішення;
- 3) передишка — період розумового відпочинку, на протязі якого винахідник відволікається від рішення поставленої задачі;
- 4) осяйнення — етап, на якому отримують нові ідеї або видозміну відомих ідей, що дозволяє отримати необхідне рішення;
- 5) доведення — етап, на якому проводиться оформлення ідеї.

Великою завадою винахідництву є психологічна інерція, яка зводиться до застосування відомих методів. Психологічна інерція — це ігнорування можливостей, крім єдиної, яка зустрічалася на початку пошуку. Психологічну інерцію підсилюють відсутність часу, страх зробити помилку тощо.

Основні методи долання психологічної інерції наступні:

1. Метод абстрагування від звичного середовища, запропонований Джоном Арнольдом. Цей метод полягає в опису середовища на уявній планеті з незвичними умовами (наприклад, сила тяжіння направлена не до центру планети, а від центра) і розробці проекту інженерних споруд на цій планеті.

2. Метод “мозкового штурму”, при якому збирають групу людей, які не зацікавлені особисто в вирішенні поставленої задачі. Кожний член колективу

пропонує свою ідею, яка записується і обробляється фахівцями. Обов'язковою умовою є відсутність критики ідеї.

3. Метод інверсії, при якій одна частина пристрою замінюється іншою. Суть інверсії в тому, щоб поміняти місцями, перевернути догори, вивернути навиворіт.

4. Метод аналогії — застосування ідеї з інших областей техніки і природи.

5. Систематичне дослідження відомих комбінацій з метою отримання нових.

При оцінці винаходу слід мати на увазі, що наукові дослідження можна поставити на службу людству тільки через винахід. Всі творіння техніки, які існують навколо нас, є впровадженими винаходами. Навіть по такому простому пристрою як олівець видано більше 20.000 патентів та авторських свідоцтв.

Винахідництво має особливість, яка відрізняє його від інших видів творчої діяльності. Відомо, що нові твори мистецтва співіснують зі старими. Нові винаходи, в силу їх конкретності та ефективності викликають необхідність заміни існуючих менш ефективних знарядь та машин. Але старе не відмирає саме по собі, а тому виникає конфлікт між винахідниками і людьми, які стоять за старими способами, машинами та теоріями. Тому винахідництво є одним з важких видів діяльності людини.

В наш час винахідництво починає оформлятися як наука.

Суттєві результати може дати систематизація винаходів, в тому числі забутих та передчасних, які випередили потреби або можливості свого часу.

1.10. Прийняття рішення

В процесі проектування значне місце займає прийняття рішення. Прийняття рішення полягає у співставленні декількох можливих можливостей або варіантів рішення і вибору найкращого з них.

Поставлена мета може бути представлена цільовою функцією типу:

$$Z = Z(X_1, X_2, \dots, X_n) \quad (1.5)$$

де X_i — незалежні параметри, які визначають характеристики системи.

Задача прийняття рішення зводиться до оптимізації цільової функції.

Якщо розглядати такі характеристики системи як надійність та вартість, то можна помітити, що зі збільшенням надійності буде збільшуватися і вартість (тут під вартістю розуміється ціна, маса, об'єм та енергоспоживання). В цьому випадку розробник стикається з необхідністю компромісу, тобто з пошуком умовного екстремуму.

Якщо рішення має кількісний вираз, то можна застосовувати кількісні методи оптимізації. Дуже часто початкові фактори мають якісний характер, що утруднює пошук оптимуму.

Прийняття рішення є процес, який характеризується наступними рисами:

1. Наявність цілі. Якщо немає цілі, немає потреби шукати рішення.

2. Наявністю декількох варіантів рішення, з якими пов'язані різні вірогідності успіху та різні витрати.

3. Необхідність врахування різних обмежень. Такими обмеженнями можуть бути габарити, форма пристрою, вага, міцність, енергоспоживання.

Блок-схема прийняття рішення представлена на рис. 1.5.

Після чіткого формулювання цілі необхідно скласти перелік можливих варіантів рішення, а також перелік обмежень. Останній використовується для скорочення варіантів. Варіанти, які залишилися, використовуються для скорочення обмежень. У підсумку можна отримати одне з чотирьох рішень:

1. Залишився один варіант. Тут існує небезпека суб'єктивного рішення, тому рекомендується повторити рішення з додавкою ще деяких варіантів.

2. Залишилося одне обмеження. Це, як правило, гарантує знаходження найкращого варіанту.

3. Не залишилося ні одного варіанту. Необхідно знайти нові.

4. Не залишилося ні одного обмеження. Вибір варіанту має деяку ступінь вірогідності.

1.11. Оцінка ефективності системи

Під ефективністю системи розуміють успішність виконання нею поставленої задачі. Це поняття дуже широке і включає в собі багато часткових характеристик, такі як надійність, точність, продуктивність тощо. Ці характеристики можна розподілити на чотири групи:

1. По продуктивності.

2. По надійності.

3. По вартості.

4. По масі та габаритам.

Продуктивність системи можна оцінювати або кількістю операцій, або вартістю готової продукції, енергії, або інформації за одиницю часу.

Надійність систем, які не відновлюються, характеризується або середнім терміном безвідмовної роботи, або вірогідністю безвідмовної роботи на протязі заданого проміжку часу.

Для систем, які можна відновлювати, необхідно враховувати додаткові показники, наприклад, коефіцієнт готовності:

$$K_z = \frac{t_{cp}}{t_{cp} + t_g}, \quad (1.6)$$

де t_{cp} — середній час роботи до відмови;

t_g — середній час пошуку та ліквідації одного пошкодження.

Це співвідношення справедливе при відсутності обмеження на кількість обслуговуючого персоналу.

Якщо кількість обслуговуючих бригад менша за кількість систем, то:

$$K_z = \frac{t_{cp} - t_n}{t_{cp} + t_g} = \frac{1 - K_n}{1 + K_g}, \quad (1.7)$$

де t_n — середній час очікування початку ремонту;

$$K_n = \frac{t_n}{t_{cp}} \text{ — коефіцієнт чекання ремонту;}$$

$$K_e = \frac{t_e}{t_{cp}} \text{ — коефіцієнт необхідності ремонту.}$$

Вартість системи визначається наступними показниками:

1. Вартістю розробки C_p , яка включає витрати на проведення НДР, проектування, виготовлення та дослідження. Вартість розробки, як правило, відносять до вартості одного (першого) зразка C_o , тобто:

$$K_p = \frac{C_p}{C_o}. \quad (1.8)$$

2. Вартість виробництва одного екземпляра системи:

$$C_n = C_o \cdot N^\mu, \quad (1.9)$$

де N — кількість пристроїв; $\mu = 0,7$.

3. Вартість експлуатації одного екземпляра системи за рік C_E визначається витратами на профілактику та ремонт, витратами на обслуговуючий персонал, енергетичними витратами. Як правило, ці витрати відносять до вартості одиничного зразка.

Слабим місцем наведених характеристик є визначення та вибір коефіцієнтів. Найчастіше вони визначаються на основі досвіду.

1.12. Технічне завдання

Вимоги до майбутньої системи управління встановлюють в документі, який називається технічним завданням (ТЗ). Основні розділи ТЗ наступні:

- підстави для розробки системи, її призначення та сфера застосування.
- умови експлуатації системи.
- експлуатаційно-технічні характеристики системи.
- об'єм робіт, що виконується виконавцем.
- технічні вимоги до системи.
- вимоги до художньо - конструкторського оформлення системи.
- вимоги до патентної захищеності системи.
- вимоги до замовника по забезпеченню розробки, впровадженню та експлуатації системи.

Розділи 7 та 8 не пов'язані з особливостями системи і мають універсальний характер. Решта розділів суттєво залежать від того, що розробляється.

В першому розділі розглядаються такі питання:

- вартість робіт, строки початку та закінчення розробки. Вказується те, чим закінчується робота (проект, дослідний екземпляр, серія і т. п.);
- перелік режимів роботи об'єкта управління, опис програми кожного режиму, правила переходу з режиму на режим, правила стиковки об'єкта з

системою. Сукупність цих даних називають алгоритмом функціонування об'єкта;

- техніко - економічне обґрунтування необхідності даної розробки (розрахунок очікуваного економічного ефекту). Наводяться дані про строки компенсації витрат;

В другому розділі висвітлюються такі питання:

- кліматичні умови експлуатації системи;
- механічні вібрації, удари і т. п., які можуть діяти на систему;
- умови живлення системи (джерело живлення, яке має замовник);
- характер роботи системи (безперервний, циклічний, одноразової дії);
- умови ремонту та догляду за системою. Наприклад, може бути вказано, що профілактичний огляд системи повинен проводитися два рази на місяць по дві години на один огляд, а ремонт — не частіше одного разу на рік строком 4 дні;

- техніко - економічні та інші наслідки відмови системи;

- засоби з техніки безпеки обслуговування системи.

В третьому розділі розглядаються питання:

- додаткові функції системи (сигналізація про пошкодження, реєстрація інформації тощо);

- комплектність, тобто склад запасних частин.

Четвертий розділ складається з таких частин:

- етапи робіт та об'єм розробки по кожному етапі, включаючи документацію на систему. При цьому вказується терміни та можливі виконавці по кожному етапу робіт;

- організації, які виготовляють зразки кожного етапу.

В п'ятому розділі розглядаються такі питання:

- інтенсивність та склад інформації системи. Кількість обслуговуючих процесів та їх перелік;

- режими роботи системи (автоматичний, напівавтоматичний, ручний, контрольний);

- потужність споживання, характер живлення, його параметри;

- технічні характеристики системи та її частин (точність, запізнення, надійність);

- вимоги до взаємозамінності блоків і необхідність застосування уніфікованих та стандартних вузлів.

Розділи 6 та 7 не потребують додаткових пояснень.

У восьмому розділі перераховуються обов'язки замовника (фінансування, проведення будівельних та монтажних робіт, підготовка персоналу).

Труднощі складання ТЗ виходять з того, що замовник мислить категоріями об'єкта управління, а виконавець використовує поняття теорії управління, тобто вони розмовляють різними технічними мовами, в яких

одне і те ж слово не завжди означає одне і теж. Виконавець, як спеціаліст в галузі автоматики, може не помітити багато “дрібниць” та тонкощів об’єкта, не врахувати того, що для замовника “зрозуміло само собою”. Замовник, добре знаючи об’єкт, не зовсім чітко уявляє можливості автоматики. Деякі, можливо другорядні, функції вимагають великої кількості апаратури, тоді як інші, складні з точки зору замовника, реалізуються досить просто.

Тому складання ТЗ потребує певного часу для уточнення, узгодження та затвердження.

Процес узгодження ТЗ містить в собі конфліктну ситуацію. Замовник прагне закласти в ТЗ найкращі характеристики, в той час як виконавець реально оцінює свої (і автоматики) можливості і хоче гарантувати виконання робіт в задані терміни.

Надзвичайно високі технічні вимоги прирікають процес проектування на провал. З іншого боку, орієнтир на відсталі рішення, відсутність елементів технічного передбачення неминуче призводять до розробки безперспективних систем.

Тому до розробки ТЗ залучаються найбільш кваліфіковані спеціалісти, що добре орієнтуються в сучасному стані систем управління.

1.13. Контрольні питання до розділу 1

1. Наведіть основні відмінності між однорівневими та багаторівневими системами управління.

2. Дайте загальну характеристику автоматизованих систем управління (АСУ).

3. Які існують етапи розвитку технічних систем?

4. Наведіть основні етапи проектування технічних систем.

5. Назвіть основні групи параметрів об’єктів проектування.

6. Як винахідництво впливає на процес проектування систем управління?

7. Як приймаються проектні рішення?

8. Критерії оцінки ефективності систем управління.

9. Які розділи входять до складу технічного завдання?

10. Як узгоджується технічне завдання?

РОЗДІЛ 2

ФОРМАЛІЗАЦІЯ ЗАВДАННЯ НА ПРОЕКТУВАННЯ ТА АЛГОРИТМУ УПРАВЛІННЯ

2.1. Умови застосування систем автоматики та управління

Розробник, перш ніж приступити до проектування системи управління, повинен вирішити важливе питання: який ступінь автоматизації необхідний на даному об'єкті. Ступінь автоматизації може змінюватися в широких межах від систем, в яких всі функції управління виконуються оператором, до систем, в яких всі ці функції виконуються автоматично. З одного боку, сучасні засоби автоматики не завжди можуть замінити оператора в системі управління. З іншого боку, існують об'єкти, управління якими оператор здійснювати не може через свої обмежені можливості. Тому найбільш розумним є сполучення оператора та автоматики в системах управління.

Умови, при яких управління за допомогою автоматики є необхідним, будуть такими:

1. Фізіологічні та психологічні можливості людини (м'язові зусилля, швидкість переробки інформації, швидкість реакції, працездатність тощо) для управління об'єктом або процесом є недостатніми.

2. Система управління знаходиться в середовищі, яке небезпечне для життя або здоров'я людини.

3. Участь людини в управлінні об'єктом вимагає від неї дуже високої кваліфікації. Підготовка таких спеціалістів майже не планується. Спеціаліст повинен мати як мінімум відповідні здібності.

4. Недостатня кількість трудових ресурсів (наприклад, в неосвоєних віддалених районах).

5. При управлінні об'єктом в критичних та аварійних ситуаціях. Спеціаліст, який добре виконує свої функції в звичайних умовах, може в аварійній ситуації бути нездібним приймати правильні рішення під впливом різних фізіологічних та психологічних факторів.

Замовник повинен описати можливі аварійні ситуації та їх наслідки. Наявність таких ситуацій може бути вирішальною при вирішенні питання про ступінь автоматизації.

Як правило, при управлінні об'єктом можна сформулювати деякий критерій якості управління. Цей критерій може мати вартісну оцінку. Замовник повинен порівняти в грошовому еквіваленті вигаши від покращання якості управління з витратами на автоматизацію (з урахуванням строків на компенсацію витрат).

2.2. Формалізація завдання на проектування

Контури майбутньої системи управління виникають у розробника не відразу. Лише після довгих розмов, суперечок, долання термінологічних бар'єрів та намагання знайти спільну мову із замовником картина починає з'ясовуватися [1, 5].

Труднощі початкового етапу роботи виникають тому, що замовник, який є спеціалістом у своїй області, мислить категоріями об'єкта управління, тоді як розробник використовує поняття, які прийняті в теорії управління. І хоч вони говорять однією мовою, бар'єр нерозуміння не зникає, тому що дуже часто одне й те ж поняття в різних галузях науки та техніки означає різні речі. Наприклад, “дросель” у електриків означає котушку з сердечником, а у теплотехніків — звуження на трубопроводі. Це примушує розробника вивчати процеси, які протікають в об'єкті управління, перш ніж він навчиться розуміти замовника з півслова. Але розробник не може бути однаковим спеціалістом і в області управління, і в технічних областях об'єкта управління. Однак, система управління повинна задовольняти вимоги замовника. Для цього розробник повинен знати про об'єкт управління не все, а те, що суттєве для процесу управління.

Розробник зображає об'єкт управління у вигляді кібернетичної “чорної скриньки”. Чорною скринькою в кібернетиці називають об'єкти, внутрішня будова яких невідома, а всю інформацію про їх структуру та функціонування можна отримати за допомогою аналізу входів та виходів.

В об'єкт завжди щось подають (матеріали або енергію) і це є входами об'єкта. Кожний технологічний процес характеризується певними технологічними параметрами (температура, тиск, стан датчиків положення тощо), які є виходами об'єкта.

Нехай об'єкт має n входів $\alpha_1, \alpha_2, \dots, \alpha_n$ та m виходів $\beta_1, \beta_2, \dots, \beta_m$ (рис. 2.1).

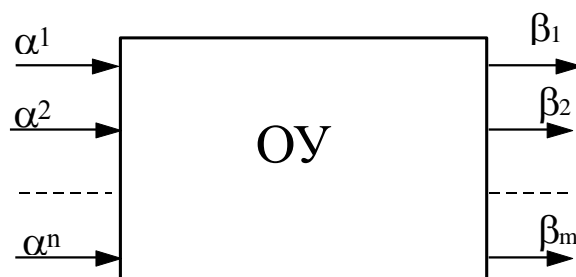


Рис. 2.1

Тепер розробник повинен в'яснити найбільш важливе питання — множину технологічних режимів, в яких може знаходитися об'єкт управління. Під технологічним режимом будемо розуміти множину обмежень на значення вихідних параметрів об'єкта.

Нехай об'єктом управління є електрична праска, в якій може змінюватися температура робочої поверхні. Вхідною величиною праски є величина струму через нагрівальний елемент. Замовник вважає, що об'єкт може мати наступні шість технологічних режимів роботи: праска виключена, перлон, штучний шовк, вовна, бавовна, льон. Кожний режим задається обмеженням на температуру робочої поверхні $t_1 \leq t \leq t_2$.

Нехай об'єктом управління є фен для сушки волосся. З точки зору замовника цей об'єкт може мати три технологічних режиму: фен виключений, включений холодний обдув, включений гарячий обдув. Вихідними параметрами, які визначають ці режими є швидкість двигуна вентилятора та температура струменя повітря. Перший режим задається парою символів $(0, V)$, де перший компонент вектора відповідає відсутності оборотів двигуна вентилятора, а другий компонент несуттєвий. Другий режим задається парою (V^*, t) , де V^* — робоче число обертів двигуна фена, а t — температура повітря в приміщенні. Третій режим задається парою (V^*, t^*) , де t^* — задана температура струменя повітря.

З принципової точки зору можливий ще один режим $(0, t^*)$, але він є неприпустимим або аварійним режимом, про що розробник повинен знати.

Якщо об'єкт управління досить складний, то його можна представити у вигляді множини більш простих об'єктів — блоків, які об'єднані функціональними та управляючими зв'язками. Такий розподіл об'єкта не є однозначним, тому що при цьому можна враховувати як конструктивний розподіл його, так і розподіл функцій, які реалізуються окремими блоками. В будь-якому випадку замовник вказує множину технологічних режимів, в яких може знаходитися кожний блок.

Сукупність режимів всіх блоків визначає режим об'єкта. Кількість можливих режимів об'єкта дорівнює кількості різних сполучень режимів окремих блоків. Ця кількість може бути дуже великою. Навіть тоді, коли блоки можуть знаходитися лише у двох режимах, при n блоках кількість режимів об'єкта дорівнює 2^n . Але більшість з цих режимів технологічно неможлива. Завдання розробника зводиться до того, щоб виділити ті режими роботи об'єкта, які можуть мати місце при його функціонуванні.

Множина допустимих режимів — перше, що суттєво для розробника при вивченні об'єктів управління. Друге, що необхідно з'ясувати, — як і при яких умовах проходить зміна режимів. У загальному випадку математична модель об'єкта має вигляд:

$$\beta_i = \phi_i(\alpha_1, \dots, \alpha_n, U_1, \dots, U_g, \delta_1, \dots, \delta_r, \beta_1, \dots, \beta_m)$$

де U_j — параметри управління; δ_k — випадкові збурення.

Список управляючих параметрів дає розробнику інформацію про те, на що може діяти система управління. Список випадкових збурень дає змогу оцінити вплив навколишнього середовища на роботу системи управління.

Після того, як об'єкт розбитий на блоки, описані всі режими блоків і розроблені методи переходу з режиму на режим, розробник до виділених блоків додає ще один, який відповідає майбутній системі управління (рис. 2.2).

Система управління на основі інформації про значення α_i , β_i , r_i (де r_i — параметри, які характеризують положення регуляторів), а також на основі знання режимів здійснює управління об'єктом. Таким чином, розробник визначає множину точок, з яких знімається інформація (датчики), та

множину точок, на які подаються управляючі дії (виконавчі механізми). Мета управління полягає або в забезпеченні можливих переходів між різними режимами, або в підтримці одного режиму. Практично між цими задачами немає суттєвої різниці, оскільки підтримання заданого режиму зводиться до організації переходів з режимів, в які може потрапити об'єкт, в заданий режим.

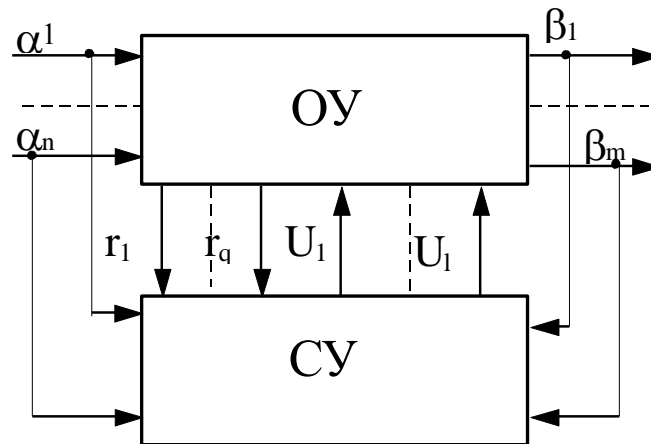


Рис. 2.2

Далі необхідно з'ясувати, як організовано при заданій множині параметрів управління перехід з одного режиму в інший за рахунок зміни значень параметрів управління. З'ясовується, куди і в якій послідовності і при яких обмеженнях у часі необхідно подавати управляючі сигнали. Виявлені послідовності подачі сигналів перероблюються в часові діаграми роботи системи управління.

Таким чином, суттєвим в об'єкті управління є:

- з яких блоків складається об'єкт з точки зору управління.
- які функціональні та управляючі зв'язки є між цими блоками та системою управління.
- які зовнішні зв'язки об'єкта і в які його блоки вони входять.
- які режими мають блоки і об'єкт в цілому і яка множина допустимих переходів між цими режимами.
- якими параметрами описуються той чи інший режими.
- які датчики та виконавчі механізми є на об'єкті.
- якими часовими діаграмами описується послідовність управляючих сигналів, щоб забезпечити той чи інший режими роботи об'єкта.

2.3. Розробка структурної схеми системи управління

З викладеного вище матеріалу впливає методика розробки структурних схем систем управління.

Після вивчення роботи об'єкта управління розробник має у своєму розпорядженні систему управління у вигляді "чорного ящика", в якому відомі її вхідні та вихідні сигнали. Внутрішній склад чорного ящика ще

невідомий і знаходження його є задачею розробника. Кожний вхід системи є вихід деякого датчика (або органа ручного управління), а кожний вихід системи є входом деякого виконавчого механізму або індикатора сигналізації, який встановлено на пульті управління.

Після цього необхідно приступити до опису алгоритму функціонування системи на формальній мові, яка дозволяє перейти до технічної реалізації системи.

Оскільки мова йде про систему управління, то не будемо розрізняти типи вхідних сигналів. Позначимо вхідні сигнали буквами X_i , а вихідні — Y_i . Змістовне значення входів та виходів відомі. Дуже часто ці сигнали є двійковими змінними, які можуть приймати лише два значення: включено-виключено (нуль та одиниця). Розробка таких систем управління і розглядається в даному посібнику.

У більшості випадків перехід від словесного опису роботи системи до опису на формальній мові неможливий через велику розмірність задачі — велику кількість вхідних та вихідних величин. Тому розробку системи починають з опису її функцій. На структурній схемі кожна з таких функцій зображується прямокутником. Для кожного прямокутника визначаються вхідні та вихідні величини, а також зв'язки між ними — алгоритм функціонування даного блоку. Крім того, необхідно визначити зв'язки між окремими блоками.

Для виконання окремих функцій можуть бути застосовані відомі пристрої або елементи. Якщо алгоритм функціонування окремого блоку дозволяє застосувати відомі методи його синтезу, то приступають до виконання цього етапу. В іншому випадку блок розбивають на частини, доки не з'явиться можливість його синтезу.

Функції окремих частин системи можуть співпадати. В цьому разі необхідно оцінити можливість та доцільність виконання їх одним і тим же пристроєм одночасно або послідовно в часі.

Дуже важливо вирішити питання про способи реалізації окремих функцій. У загальному випадку вони можуть бути реалізовані апаратними або програмними засобами. Перші, як правило, мають меншу величину запізнення, але і мають жорсткий алгоритм функціонування та вимагають великої кількості елементів. Програмні засоби, особливо при застосуванні мікропроцесорів, можуть мати гнучкий алгоритм функціонування, але характеризуються більшою величиною затримки. При управлінні технологічними процесами затримки, як правило, набагато менші за критичні. При обробці інформації (наприклад, при визначенні спектральних характеристик високочастотних сигналів в реальному масштабі часу) ці затримки можуть бути занадто великими.

Частини функціональної схеми повинні працювати узгоджено, що досягається синхронізацією виконання функцій. Синхронізаторами можуть бути генератор тактових імпульсів або інші пристрої.

Подібний метод розробки структурної або функціональної схеми застосовують тоді, коли математична модель системи управління має алгоритмічний характер. Якщо система управління описується системою певних рівнянь, то структура системи визначається цими рівняннями. Такий підхід має місце, наприклад, в системах телемеханіки, в яких структурна схема кодера (декодера) базується на рівняннях побудови коду інформації.

В системах автоматичного регулювання структурну схему не розробляють, а вибирають з невеликої кількості уже розроблених. Як правило, структурна схема системи регулювання складається з об'єкта, датчика та задатчика параметрів, регулятора та виконавчого механізму. Для цих елементів необхідно визначити (шляхом експерименту або з літературних джерел) передаточні функції, за якими визначають передаточну функцію всієї системи. Замість розробки структурної схеми, в цьому разі необхідно визначити характер перехідних процесів в системі і їх відповідність вимогам технічного завдання. Зближення показників перехідних процесів системи і вимог до неї здійснюється шляхом вибору закону регулювання і розрахунків положення відповідних елементів регулятора.

Розробка структурної схеми є неформалізованою задачею і може бути викладена лише у загальному вигляді.

Одна і та ж система може мати декілька варіантів рішення, декілька структурних схем. Тому необхідно провести їх аналіз і вибрати найбільш оптимальний варіант з точки зору ефективності, апаратурних витрат та інших показників технічного завдання.

Для кожного блоку структурної схеми необхідно визначити вимоги до структури і форми вхідних та вихідних сигналів. Це дає змогу при відомому алгоритмі функціонування провести його синтез.

2.4. Формалізація алгоритму управління

Для розробки системи управління з мінімальною кількістю елементів необхідно описати закон її функціонування на формальній мові, яка дозволяє перейти від словесного опису алгоритму управління до її структурної або функціональної схеми. Найчастіше це математична форма. Тому формалізація – це перехід від словесного опису алгоритму управління до математичного. Математична форма опису дозволяє перейти до структурної (функціональної) схеми системи управління і при необхідності розробити блок-схему алгоритмів управління.

Для формалізації алгоритму управління механізмами технологічної лінії можливе застосування двох методів.

Перший з них, більш універсальний, заснований на представленні математичної моделі управління механізмом у вигляді *RS*-тригера, в якому сигнал на *S*- вході включає тригер (механізм), а сигнал на *R*-вході – виключає його [3]. Кожне рівняння визначає сигнал на *S*- або на *R*-вході тригера. Це булеві рівняння, значення яких може дорівнювати одиниці або нулю. Якщо

дорівнює одиниці S -рівняння, то механізм буде включено. Якщо дорівнює одиниці R -рівняння, то механізм буде виключено. Якщо ці рівняння дорівнюють нулю, то стан механізму не змінюється. Логіка роботи механізмів забезпечує умови, коли комбінація $S = R = 1$ неможлива. Розподіл умов включення та виключення механізму спрощує процес складання таких рівнянь.

Умови роботи механізмів, включають положення технологічного матеріалу, положення робочих органів механізмів та стан механізмів (включено/виключено). Такі змінні утворюють кон'юнкцію і на структурній схемі об'єднуються елементами "І". В рівняння управління можуть входити додаткові тригери, які запам'ятовують стан механізмів в попередні моменти часу. Такі тригери відіграють роль датчиків, установка яких неможлива або небажана. Їх можна назвати математичними датчиками. Положення технологічного матеріалу або робочих органів механізмів визначаються відповідними датчиками положення. Стан механізму можна визначити по стану його пускового пристрою, які часто мають для цього спеціальний контакт (блок-контакт).

В технологічних лініях можуть застосовуватися різні механізми. З точки зору управління ця різноманітність може бути зведена до трьох типів:

- механізми безперервного руху;
- крокові механізми;
- човникові механізми.

Механізми безперервного руху не мають датчиків положення, тому для управління ними необхідні зовнішні команди на включення та на виключення.

Крокові механізми переміщують свої робочі органи на крок, величина якого задається датчиком (датчиками), причому кількість кроків в одному напрямку необмежена. Величина кроку може бути постійною або змінною, що визначається конструкцією датчика.

Крокові механізми включаються зовнішньою командою, а зупиняються під дією свого датчика кроку. Особливість цих механізмів полягає в тому, що початок і кінець їх руху відбувається при одному і тому ж стані датчика. Тому на початку руху необхідно нейтралізувати дію датчика, що краще за все зробити за допомогою додаткового тригера.

Човникові механізми – це реверсивні механізми, які виконавши рух в одному напрямку, повинні повернутися в початкове положення. Крайні положення механізму визначаються двома датчиками (інколи одним, спеціальної конструкції, який переключається в крайніх положеннях механізму).

В [9] наведені базові рівняння управління всіма типами механізмів. В ці рівняння необхідно підставити конкретні змінні для заданої технологічної ділянки. В автоматичному режимі управління це сигнали датчиків положення технологічного матеріалу, датчиків положення механізмів, стан механізмів

(стан тригерів управління) та стан додаткових тригерів. Для ручного управління в ці рівняння можуть входити кнопки “ПУСК” та “СТОП” і, можливо, диспетчерські ключі, які визначають ручний або автоматичний режими управління механізмами.

Розглянемо приклад складання рівнянь управління за даною методикою.

Нехай необхідно розробити систему управління технологічною ділянкою для вибивки ливарних форм. Мнемосхема технологічної ділянки наведена на рис. 2.3. Механізми цієї ділянки повинні працювати за таким алгоритмом.

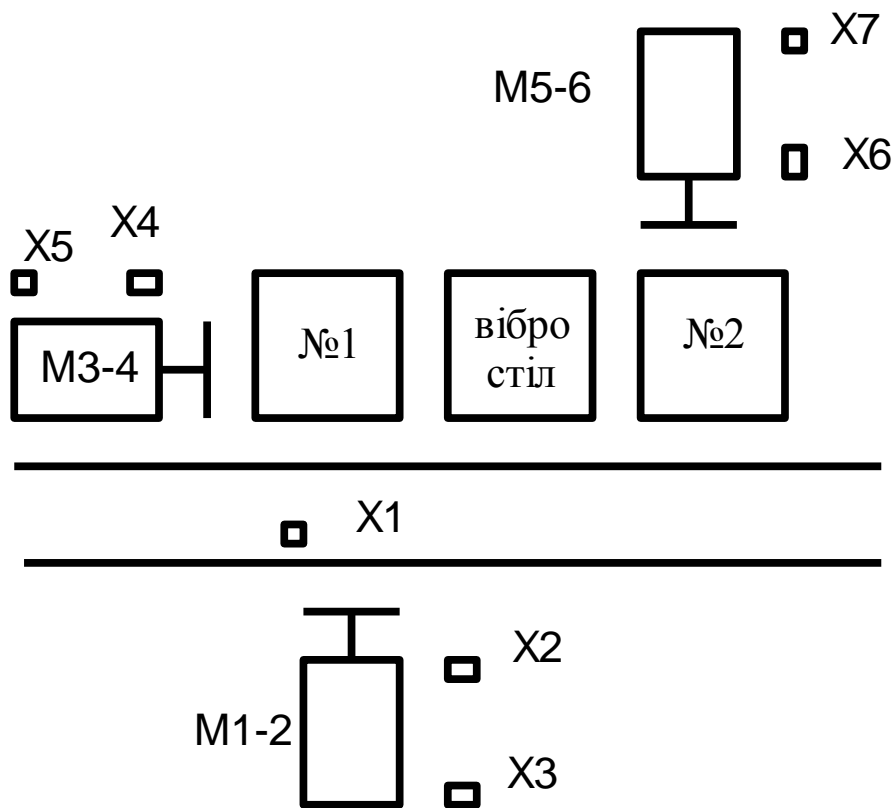


Рис. 2.3. Технологічна ділянка вибивки ливарних форм

Форми з відлитими деталями переміщуються транспортером, який працює безперервно, а тому він не входить в систему управління. Коли форма опиниться в зоні дії реверсивного механізму $M1-2$ ($X1 = 1$), останній повинен перемістити її на проміжний стіл №1 і повернутися в початкове положення. Коли форма опиниться на проміжному столі, а механізм $M1-2$ повернеться в початкове положення, реверсивний механізм $M3-4$ повинен перемістити її на вібростіл і повернутися в початкове положення. На вібростолі форма звільняється від суміші, яка висипається з форми. Разом з сумішшю випадає і відлита деталь.

Наступна форма буде переміщена на проміжний стіл №1. Механізм $M3-4$, переміщуючи наступну форму на вібростіл, перемістить попередню на

проміжний стіл №2. Після цього реверсивний механізм $M5-6$ повинен перемістити порожню форму на транспортер.

Дозволяється одночасна робота механізмів $M1-2$ та $M5-6$. Решта механізмів одночасно працювати не можуть.

Датчик $X1$ визначає положення форми. Решта датчиків визначають положення механізмів.

Згідно діючих правил реверсивні механізми позначаються двома цифрами. Реверсивні механізми – це механізми, які можуть переміщуватися (обертатися) в двох напрямках. Для цього вони обов'язково мають два пускових пристрої, один з яких включає механізм для руху в одному напрямку, другий – в іншому. З точки зору управління, такі механізми розглядаються як два нереверсивні. Тому при складанні рівнянь управління кожний реверсивний механізм “розпадається” на два. Так, наприклад, механізм $M1-2$ розглядається як механізм $M1$ та $M2$.

За зовнішнім виглядом реверсивний механізм може нічим не відрізнятися від нереверсивного. Наприклад, напрям оберту електричного трифазного двигуна з короткозамкненим ротором залежить від того, як підключаються його статорні обмотки до мережі живлення. Якщо поставити два контактора, кожний з яких по різному підключає обмотки до мережі, то можна управляти напрямом руху механізму. В електричних двигунах одночасне включення обох контакторів неприпустиме. Ротор двигуна обертатися не буде, а тому індуктивний опір обмоток дуже малий. В такому режимі через обмотки проходить дуже великий струм і вони згорять.

Гідравлічні та пневматичні механізми не бояться такого режиму. Одночасна подача масла (повітря) в обидві порожнечі механізму (ліворуч та праворуч від поршня) не дасть змоги поршню рухатися, але з ладу його не виведе.

Складемо рівняння управління механізмом $M1$, який включаємо, якщо:

- є форма на транспортері ($X1 = 1$);
- механізм $M2$ не включений ($M2 = 0$);
- механізм $M3$ знаходиться в лівому крайньому положенні ($X5 = 1$).

Тоді рівняння включення механізму $M1$ має вигляд:

$$M1_S = X1 \cdot \overline{M2} \cdot X5. \quad (2.1)$$

Механізм $M1$ виключається в своєму крайньому положенні датчиком $X2$:

$$M1_R = X2. \quad (2.2)$$

Механізм $M2$ повинен включитися, коли механізм $M1$ знаходиться в крайньому положенні ($X2 = 1$) і буде виключений ($M1 = 0$):

$$M2_S = X2 \cdot \overline{M1}. \quad (2.3)$$

Виключається механізм $M2$ тоді, коли він повернеться в початкове положення ($X3 = 1$):

$$M2_R = X3. \quad (2.4)$$

Механізм $M3$ повинен перемістити форму на вібростіл. Але датчика, який би вказував на наявність форми на проміжному столі №1, немає. Датчик наявності форми на столі №1 поставити можна, але це знижує надійність системи управління. Створимо його штучно. Для цього введемо тригер T , який включиться тоді, коли працював механізм $M1$ (без форми він не включається). Таким чином, включений стан тригера T говорить про наявність форми на проміжному столі №1. Рівняння включення тригера має вигляд:

$$T_S = M1. \quad (2.5)$$

Тригер T необхідно виключити, коли механізм $M4$ повертається в початкове положення:

$$T_R = M4. \quad (2.6)$$

Тепер можна скласти рівняння управління механізмом $M3$. Його можна включити, якщо:

- є форма на проміжному столі ($T = 1$);
- механізм $M1$ знаходиться в початковому положенні ($X3 = 1$);
- не працює механізм $M4$ ($M4 = 0$).

Рівняння включення механізму $M3$ має вигляд:

$$M3_S = T \cdot X3 \cdot \overline{M4}. \quad (2.7)$$

Механізм $M3$ виключається датчиком положення $X4$ в крайньому положенні:

$$M3_R = X4. \quad (2.8)$$

Механізм $M4$ включається тоді, коли механізм $M3$ займає крайнє положення ($X4 = 1$) і виключений ($M3 = 0$):

$$M4_S = X4 \cdot \overline{M3}. \quad (2.9)$$

Виключається механізм $M4$ датчиком $X5$ в початковому положенні:

$$M4_R = X5. \quad (2.10)$$

Механізм $M5$ може працювати одночасно з механізмом $M1$. При цьому його перше переміщення буде “холостим”, тому що в його зоні ще немає форми. Але це не має принципового значення з точки зору технології. По закінченні роботи лінії на вібростолі залишиться одна форма, що теж не має значення.

Таким чином, механізм $M5$ можна включити при включенні механізму $M1$, але при виключеному механізмі $M6$:

$$M5_S = M1 \cdot \overline{M6}. \quad (2.11)$$

Виключається механізм $M5$ датчиком положення $X6$ в своєму крайньому положенні:

$$M5_R = X6. \quad (2.12)$$

Механізм $M6$ включається тим же датчиком $X6$, коли буде виключено механізм $M5$:

$$M6_S = X6 \cdot \overline{M5}. \quad (2.13)$$

Виключається механізм M_6 датчиком X_7 в початковому положенні:

$$M_{6R} = X_7. \quad (2.14)$$

Система рівнянь (2.1 ÷ 2.14) і є математичною моделлю алгоритму управління механізмами заданої технологічної лінії.

Питання про те, як від цієї системи рівнянь перейти до структурної схеми, буде розглянуто далі.

Другий метод формалізації алгоритму управління полягає в складанні логічної схеми алгоритмів (ЛСА) [4]. Цей метод можна застосовувати тоді, коли механізми працюють в певній послідовності і коли механізми не працюють одночасно. Це можливо на невеликих технологічних ділянках. На великих технологічних лініях механізми можуть працювати одночасно. В цьому разі застосовувати ЛСА недоцільно.

ЛСА є своєрідною формою запису програми роботи механізмів (програма записана в псевдокодах). Вона складається з виконання команд включення/виключення механізмів та перевірки положення механізмів або технологічного матеріалу. Всі оператори ЛСА записуються в рядок зліва направо, в порядку їх виконання. Якщо в ЛСА записаний логічний оператор (перевірка стану датчика), то праворуч нього ставиться стрілка, вістря якої направлене вгору. Ця стрілка повинна мати довільний, але конкретний, номер. Праворуч логічного оператора (X_i) записується той наступний оператор, який виконується тоді, коли $X_i = 1$. Якщо $X_i = 0$, то управління передається іншому операторові ЛСА, перед яким ставиться стрілка, вістря якої направлене вниз і яка має той же номер, що й стрілка за X_i .

Для складання ЛСА необхідно позначити всі команди на включення та виключення механізмів великими латинськими літерами, або однією літерою з індексом. Це позначення може бути довільним, але найкраще, коли позначення в ЛСА співпадає з позначенням механізмів на мнемосхемі. Доцільно також позначати команди на включення механізмів M_i , а команди на виключення – M_i^* .

Аналогічно позначаються і датчики. Необхідно лише вказати, при яких умовах $X_i = 1$, а при яких – $X_i = 0$.

Якщо завдання зрозуміле, то складання ЛСА не є складною задачею. Її треба записати таким чином, щоб виконувався заданий алгоритм роботи всіх механізмів.

Як правило, при такому підході в ЛСА є “зайві” оператори. Тому ЛСА необхідно спростити. Це спрощення виконується в такій послідовності:

- в ЛСА об’єднуються і замінюються однією ті команди, які в усій ЛСА записані поруч (однією командою можна включити або виключити декілька механізмів);

- зменшення кількості команд за рахунок так званих очікуючих команд. Очікуючі команди – це команди типу:

$$\downarrow_n M_i X_k \uparrow^n M_i^*. \quad (2.15)$$

В такій частині ЛСА команда M_i^* виконується лише тоді, коли X_k стане дорівнювати одиниці, тобто тоді, коли M_i виконав свою дію. В цьому разі команду M_i^* можна з ЛСА вилучити, при умові, що команда M_i не буде запам'ятовуватися. Для систем управління, які реалізуються на базі мікропроцесора, таке скорочення ЛСА неприпустиме, тому що вихідні порти мікропроцесора мають тригери, а тому команда M_i запам'ятовується.

Подальше скорочення ЛСА можливо за рахунок усунення повторень одних і тих же операторів. Ця процедура вимагає застосування формул переходів і детально викладена в [4].

В спрощеній ЛСА оператори об'єднують в мікрокоманди. Цю операцію виконують лише тоді, коли система управління розробляється на базі мікропрограмного автомату. В цьому випадку цей розділ складається з двох підрозділів: складання ЛСА та формування мікрокоманд.

Наведемо приклад розробки ЛСА та формування мікрокоманд для тієї ж ділянки для вибивки ливарних форм, що наведена на рисунку 5.1.

Проведемо змістовний опис операторів:

M_i – команда на включення i -го механізму;

M_i^* – команда на виключення i -го механізму;

$X_i = 1$ – i -тий механізм знаходиться в контрольованій точці;

$X_i = 0$ – i -тий механізм не знаходиться в контрольованій точці.

ЛСА процесу управління має вигляд:

$$\begin{aligned}
 U = & A_0 \downarrow_1 X1 \uparrow^1 \downarrow_2 M1 \cdot X2 \uparrow^2 M1^* \downarrow_3 M2 \cdot X3 \uparrow^3 M2^* \\
 & \downarrow_4 M3 \cdot X4 \uparrow^4 M3^* \downarrow_5 M4 \cdot X5 \uparrow^5 M4^* \\
 & \downarrow_6 M5 \cdot X6 \uparrow^6 M5^* \downarrow_7 M6 \cdot X7 \uparrow^7 M6^* \omega \uparrow^1
 \end{aligned} \tag{2.16}$$

В цій ЛСА є очікуючі оператори типу $\downarrow_n M_I X_J \uparrow^n M_I^*$. Відомо, що в таких структурах можна вилучити M_i^* при умові, що команда M_i запам'ятовуватися не буде.

Нова ЛСА має вигляд:

$$\begin{aligned}
 U = & A_0 \downarrow_1 X1 \uparrow^1 \downarrow_2 M1 \cdot X2 \uparrow^2 \downarrow_3 M2 \cdot X3 \uparrow^3 \downarrow_4 M3 \cdot X4 \uparrow^4 \\
 & \downarrow_5 M4 \cdot X5 \uparrow^5 \downarrow_6 M5 \cdot X6 \uparrow^6 \downarrow_7 M6 \cdot X7 \uparrow^7 \omega \uparrow^1
 \end{aligned} \tag{2.17}$$

Подальше скорочення ЛСА неможливе. Перейдемо до формування мікрокоманд. Для цього визначимо сумісність операторів. Сумісність функціональних операторів (команд на включення та виключення механізмів) зображено на рис. 2.4.

Ця таблиця заповнюється таким чином: механізми $M1$ та $M2$ є частинами одного і того ж реверсивного механізму $M1-2$. Їх одночасне включення недопустиме, а тому на перехрещенні стовпця $M1$ та рядка $M2$ стоїть знак “ X ” – заборона одночасного включення. Те ж саме можна сказати про механізми $M3$ та $M4$, $M5$ та $M6$.

<i>M2</i>	X				
<i>M3</i>	X	X			
<i>M4</i>	V	V	X		
<i>M5</i>	V	V	V	V	
<i>M6</i>	V	V	V	V	X
	<i>M1</i>	<i>M2</i>	<i>M3</i>	<i>M4</i>	<i>M5</i>

X – включати одночасно механізми заборонено;
V – включати одночасно механізми можна.

Рис. 2.4. Сумісність функціональних операторів

Механізми *M1* та *M3* одночасно включати не можна, тому що траєкторії їх руху перехрещуються. Те ж саме можна сказати про механізми *M2* та *M3*.

Функціональний та логічний оператори несумісні тоді, коли останній змінює свій стан під дією першого.

M1 – (X_1, X_2, X_3); *M2* – (X_2, X_3); *M3* – (X_4, X_5); *M4* – (X_4, X_5);
M5 – (X_6, X_7); *M6* – (X_6, X_7).

Визначимо сумісність функціональних та логічних операторів:

Очікуючі оператори завжди несумісні. Але якщо поміняти місцями функціональний та логічний оператори, то вони стають сумісними [4].

Побудуємо першу мікрокоманду (рис. 2.5):

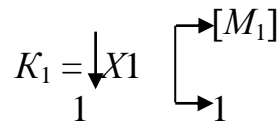


Рис. 2.5. Перша мікрокоманда

Ця команда побудована таким чином: ЛСА (5, 17) першим оператором є оператор $X1$ (A_0 – не враховується). З нього і починається команда K_1 . Оскільки це логічний оператор, від нього ведемо дві стрілки: верхня відповідає $X1=1$, нижня $X1=0$. При $X1=0$ вістря стрілки 1, в ЛСА знову веде до $X1$, тобто ніяких дій не відбувається. При $X1=1$ повинен бути включений оператор $M1$. Але $M1$ та $X1$ несумісні. Тому ця гілка закінчується. Розміщення $M1$ в дужках означає, що оператор $M1$ не входить до команди $K1$. Нова команда $K2$ повинна починатися з оператора $M1$. Але оператори $M1$ та $X2$ є очікуючі. Їх можна поміняти місцями, але гілка $X2=0$ закінчується оператором $M1$ [9]. Після $X2=1$ повинен виконуватися оператор $M2$, а за ним оператор – $X3$. Ця пара операторів теж є очікуючою парою, тому їх теж міняємо місцями. Подальша побудова команди $K2$ аналогічна.

Побудуємо другу мікрокоманду (рис. 2.6).

В результаті маємо дві команди, які включають всі логічні та функціональні оператори ЛСА.

Питання про те, як по цим мікрокомандам побудувати структурну схему системи управління, буде розглянуто далі.

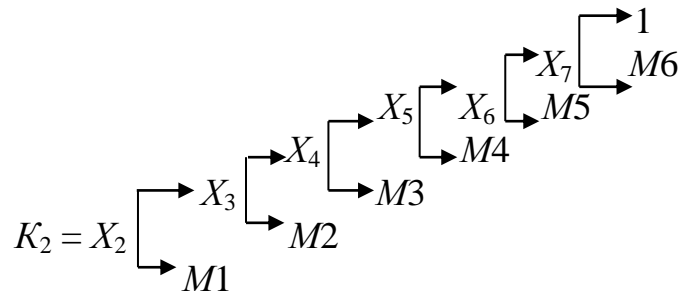


Рис. 2.6. Друга мікрокоманда

Між роботою механізмів по рівняннях (5.1 ÷ 5.14) та по ЛСА (5.17) є суттєва різниця. При управлінні по рівняннях (апаратним або програмним способом) система управління весь час визначає стан датчиків і виробляє відповідні сигнали управління механізмами. При цьому допускається одночасна робота механізмів (якщо це обмеження не закладено в алгоритм). Так можуть працювати одночасно механізми $M1-2$ та $M5-6$. Час повного циклу управління складається з часу переміщення механізмів $M1, M2, M3$ та $M4$, тобто дорівнює $4T$ (де T – час переміщення одного механізму).

При роботі системи по ЛСА управління механізмами виконується послідовно в часі. Система управління, включивши механізм, чекає переключення відповідного датчика і тільки тоді переходить до управління наступним механізмом. В цьому випадку час всього циклу управління дорівнює $6T$. В принципі, при роботі по рівняннях є можливість збільшити швидкість подачі форм за рахунок збільшення швидкості транспортера. Чи дійсно це можна зробити, залежить від інших ділянок повної технологічної лінії.

Якщо система управління реалізується на мікропроцесорі або мікроконтролері, то формалізація алгоритму роботи можна проводити і у вигляді рівнянь управління, і у вигляді ЛСА.

2.5. Контрольні питання до розділу 2

1. Які основні причини застосування систем управління на промислових підприємствах?
2. Наведіть основні математичні моделі об'єктів проектування.
3. Як взаємодіють об'єкту управління та система управління?
4. Методи розробки структурних схем систем управління.
5. Наведіть методику складання рівнянь управління.
6. Які типи механізмів розглядаються при складанні рівнянь управління?
7. Як скласти логічну схему алгоритму (ЛСА)?
8. Що таке сумісність функціональних та логічних операторів?
9. Як побудувати мікрокоманди за ЛСА?

РОЗДІЛ 3 ПРОЕКТУВАННЯ СКІНЧЕНИХ АВТОМАТІВ

3.1. Загальні відомості про скінчені автомати

Скінченим автоматом називається дискретний пристрій, вихідний сигнал якого визначається не тільки вхідним набором в поточний момент часу, але й вхідними наборами в попередні моменти часу. Це можливо при наявності в структурі автомата елементів пам'яті, тому такі автомати називають ще автоматами з пам'яттю. Елемент пам'яті — це дискретний елемент, вихід якого може бути нульовим або одиничним. Комбінація виходів елементів пам'яті називається внутрішнім станом автомата або просто станом.

Це визначення є ключовим в розумінні суті скінченого автомату [7, 10 – 14].

Автомат може знаходитися в стійкому або нестійкому стані. Стійким називається стан, який може змінюватися лише при зміні вхідних наборів. Нестійкий стан змінюється без зміни вхідних наборів, що пов'язано з перехідними процесами в схемі. Перехід з одного стійкого стану в інший стійкий здійснюється лише через нестійкий.

При теоретичному дослідженні такий автомат розглядається як пристрій з одним входом та одним виходом. Теоретичний або абстрактний автомат є математичною моделлю реального автомата.

У загальному випадку робота автомата описується двома системами рівнянь:

$$Z_{i(t+1)} = f(X_t, Z_{it}); \quad (3.1)$$

$$Y_j = f(X_t, Z_{it}). \quad (3.2)$$

Рівняння (3.1) називаються рівняннями переходів. Вони показують, як змінюється внутрішній стан автомата в залежності від вхідного набору X_t та стану автомата в даний момент часу.

Рівняння (3.2) називають рівняннями виходів. Вони показують зміну виходів в залежності від вхідного набору та стану автомата.

Автомати з рівнянням (3.2) називаються автоматами Міллі. Існують автомати з рівняннями виходів типу:

$$Y_j = f(Z_{it}).$$

Такі автомати називаються автоматами Мура. Існує велика кількість літератури, в якій наведена методика переходу від одного типу автомата до іншого. Це чисто теоретичне питання. Розробнику, а тим більше замовнику, байдуже, який тип автомата у нього вийде.

Автомати можуть бути синхронні та асинхронні. В синхронних автоматах перехід з одного стану в інший може здійснюватися лише в момент подачі синхроімпульсів, які виробляються спеціальним генератором. В асинхронних автоматах перехід з одного стану в інший здійснюється тільки під впливом зміни вхідного набору.

Задача проектування скінченного автомата зводиться до знаходження його структури — типів елементів, їх кількості та способів зв'язку між собою, — яка забезпечує виконання ним заданого алгоритму роботи. Цей алгоритм задається в словесній формі і перехід від цієї форми до конкретних рівнянь (3.1) та (3.2) є основною задачею проектування скінчених автоматів.

Словесний опис алгоритму роботи автомата переводиться в таблицю переходів та таблицю виходів. Така форма зображення роботи автомата зручна тоді, коли вхідні сигнали надходять на вхід автомата у вигляді паралельного коду. Для автоматів, у яких вхідні набори надходять у вигляді послідовного коду, на початковому етапі застосовують граф.

Синтез автомата проводять в два етапи. На першому етапі розглядається абстрактний автомат, в якому не беруться до уваги фізичні принципи представлення вхідних та вихідних сигналів, а також елементів, з яких він буде складатися.

На другому етапі проектується структурний автомат, в якому враховуються реальні характеристики сигналів та елементів.

Рівняння, які описують роботу автомата, мають три невідомих. Для рішення цих рівнянь задаються значенням Z_{it} . Ця процедура називається кодуванням стану автомата і значною мірою визначає його структуру.

Структурна схема скінченного автомата в загальному випадку має вигляд, наведений на рис. 3.1. Основу автомата складає комбінаційна схема, на вхід якої поступають вхідні сигнали X та виходи елементів пам'яті Z . Ця схема формує вихідні сигнали автомата Y та сигнали управління елементами пам'яті Q .

3.2. Проектування автоматів з паралельними входами

3.2.1. Проектування автоматів на зворотних зв'язках

Зворотний зв'язок з елементами затримки є одним з типів елементів пам'яті, які можуть застосовуватися в автоматах.

Синтез автоматів проводиться в такій послідовності [10 – 14]:

- складання первинної таблиці переходів або таблиці переходів абстрактного автомата;
- мінімізація таблиці переходів автомата або мінімізація його станів;
- кодування станів автомата;
- складання таблиці переходів структурного автомата і отримання рівнянь переходів;
- складання таблиці виходів структурного автомата і отримання рівнянь виходів;
- побудова структурної схеми автомата;
- вибір елементів і перехід до принципової схеми.

В цій послідовності етап складання таблиці переходів абстрактного автомата є найважчим і найголовнішим, тому що він носить неформальний характер.

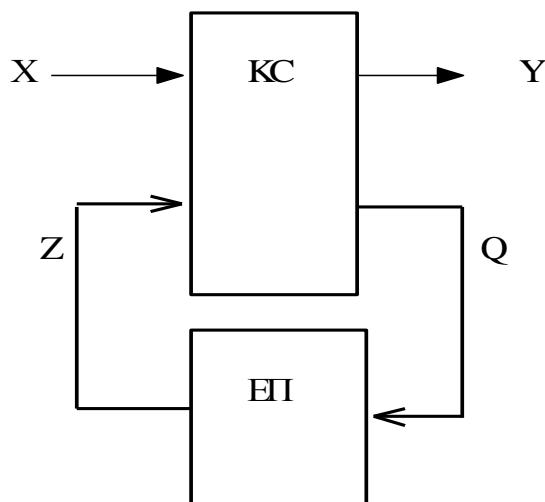


Рис. 3.1

Таблиця переходів абстрактного автомата або таблиця Хаффмена має 2^n стовпців (де n — кількість вхідних змінних). Крім того, для кожного виходу автомата вводять ще по одному стовпцю. Кількість рядків таблиці переходів визначити наперед неможливо.

Для кожного вхідного набору за алгоритмом роботи автомата необхідно визначити можливі його стійкі стани. Їх кількість визначається кількістю попередньої інформації, яку необхідно запам'ятати. Для кожного стану відводиться свій рядок в таблиці переходів. Таким чином, в рядку може бути лише один стійкий стан, що дозволяє контролювати правильність складання таблиці. Позначення стану може бути будь-яким: буквою, цифрою тощо. Як правило, це позначення беруть в дужки, щоб відрізнити стійкий стан від нестійкого.

Решта клітинок таблиці переходів заповнюється нестійкими станами або ризиками, якщо даний вхідний набір неможливий.

Заповнення клітинок таблиці переходів нестійкими станами проводиться таким чином: беремо за початковий який-небудь стійкий стан і змінюємо вхідний набір, тобто переміщуємося в таблиці в горизонтальному напрямку. При такому вхідному наборі автомат повинен мати певне значення виходу, якому відповідає певний стійкий стан в цьому стовпчику. Тому в даній клітині ставимо такий же стан, але нестійкий. Якщо перехід по горизонталі з будь-яких причин неможливий, то в даній клітині ставимо ризику. Повертаємося в початковий стан і повторюємо цю процедуру для іншого вхідного набору, доки не будуть заповнені всі клітини таблиці переходів. Оскільки процес складання таблиці переходів визначається конкретним алгоритмом роботи автомата, описати цю процедуру в загальному вигляді неможливо. Тому в існуючій літературі таблиці

переходів автомата наводяться без пояснення процесу їх побудови. Процес складання таблиці переходів абстрактного автомата можна пояснити на прикладі.

Нехай необхідно розробити автомат з двома входами і одним виходом, який працював би за таким алгоритмом:

1. Якщо $X_2 = 1$, то $Y = X_1$;
2. Якщо $X_2 = 0$, то $Y_t = Y_{t-1}$.

Одночасно може змінюватися лише одна змінна. Це обмеження не є загальним. Його може і не бути. В словесній формі цей алгоритм описується наступним чином: якщо на другому вході автомата сигнал дорівнює одиниці, то вихід приймає значення першого входу; якщо на другому вході сигнал дорівнює нулю, то вихід зберігає попереднє значення.

Дві вхідні змінні та один вихід визначають 5 стовпців таблиці переходів (табл. 3.1).

Таблиця 3.1

		X_1X_2				
		00	01	11	10	Y
[1]	3	—	5	0		
[2]	3	—	6	1		
1	[3]	4	—	0		
—	3	[4]	6	1		
1	—	4	[5]	0		
2	—	4	[6]	1		

Заповнення таблиці виконано таким чином. При $X_1 = 0$, $X_2 = 0$, згідно з алгоритмом, вихід повинен зберігати своє попереднє значення. Це попереднє значення може бути як нульовим, так і одиничним. Тому відведемо для цієї вхідної комбінації два стійких стана — [1], з нульовим виходом автомата, та [2] — з одиничним виходом.

При вхідному наборові $X_1 = 0$, $X_2 = 1$, вихід автомата повинен дорівнювати значенню X_1 , тобто дорівнювати нулю. Відведемо для цієї вхідної комбінації стійкий стан [3] з нульовим виходом.

При $X_1 = 1$, $X_2 = 1$, вихід автомата повинен дорівнювати значенню X_1 , тобто одиниці. Відведемо для цієї вхідної комбінації стійкий стан [4] з одиничним виходом.

При $X_1 = 1$, $X_2 = 0$, вихід автомата зберігає попереднє значення, яке може бути як нульовим, так і одиничним. Тому відведемо для цієї вхідної комбінації два стійких стана: [5], з нульовим виходом, та [6] — з одиничним.

Решту клітин таблиці переходів заповнюємо таким чином. Візьмемо, як початковий, стійкий стан [1]. Змінимо вхідну комбінацію з 00 на 01. Згідно з алгоритмом роботи автомата, його вихід зберігає попередній, тобто нульовий, стан, чому відповідає стійкий стан [3]. Тому в цій клітині поставимо нестійкий стан 3. Зміна вхідної комбінації з 00 на 11 неможлива,

тому що при цьому змінюється стан двох вхідних змінних. Тому в даній клітині ставимо ризику. При зміні вхідної комбінації з 00 на 10 вихід автомата повинен зберігати своє попереднє, тобто нульове значення, чому відповідає стійкий стан [5]. Тому в даній клітині ставимо нестійкий стан 5.

Решта клітин таблиці переходів заповнюються аналогічно.

Таким чином, отримали таблицю переходів, яка має шість рядків. Кількість рядків таблиці переходів визначає і кількість елементів пам'яті, які необхідні для реалізації заданого алгоритму. Співвідношення між ними виражається формулою:

$$2^z \geq N,$$

де Z — мінімальна кількість елементів пам'яті, яка задовольняє цю нерівність; N — кількість рядків таблиці переходів.

Для даного автомата необхідно мати три елементи пам'яті. Кількість елементів пам'яті визначає складність автомата, тому що для управління цими елементами необхідно мати відповідну схему. Тому необхідно визначити можливість скорочення кількості елементів пам'яті. Це можливо, якщо автомат має еквівалентні стани. Еквівалентними називаються стани, при яких автомат однаково реагує на вхідні сигнали. Два стани — два рядки таблиці переходів — еквівалентні, якщо в кожній клітині їх однакові елементи, причому стійкі та нестійкі стани з однаковими номерами (з однаковими позначеннями) вважаються еквівалентними. Клітину з ризику можна віднести до будь-якого стану.

В таблиці 3.1 однаковими (еквівалентними) будуть рядки 1, 3 та 5, а також 2, 4 та 6.

Еквівалентні рядки об'єднуються за таким правилом: стійкий стан з нестійким або ризику утворює стійкий стан; нестійкий стан з нестійким або ризику утворює нестійкий стан; ризику з ризикуми утворюють ризику.

Спрощена таким чином таблиця переходів має вигляд таблиці 3.2.

Таблиця 3.2

		X_1X_2				
		00	01	11	10	Y
Z_i	0	[1]	[3]	4	[5]	0
	1	[2]	3	[4]	[6]	1

Таким чином, шість рядків таблиці переходів зведені до двох рядків. Для реалізації такого автомата досить одного елемента пам'яті. Стан цього елемента може бути нульовим або одиничним. Тому закодуємо рядки таблиці цими станами. В разі одного елемента пам'яті це кодування може бути довільним. Нехай 1-й рядок буде мати код нуль, а другий — одиницю.

На цьому закінчується проектування абстрактного автомата і необхідно перейти до проектування структурного автомата.

Таблиця переходів структурного автомата має 2^n стовпців, кожний з яких відповідає певному вхідному наборові, та таку ж кількість рядків, яку має таблиця переходів абстрактного автомата. Елементами таблиці буде стан автомата в наступний момент часу, тобто Z_{t+1} .

Таблицю переходів структурного автомата отримують з таблиці переходів абстрактного автомата за таким формальним алгоритмом: на місце стійких станів в таблиці переходів абстрактного автомата ставимо код цього рядка; на місце нестійкого стану ставимо код рядка, який відповідає такому ж стійкому стану.

Таблиця переходів структурного автомата, отримана з таблиці 3.2, має вигляд таблиці 3.3.

Таблиця 3.3

		X_1X_2			
		00	01	11	10
Z_t	0	0	0	1	0
	1	1	0	1	1
		Z_{t+1}			

Ця таблиця є не що інше, як карта Карно рівняння переходів. Тому:

$$Z_{t+1} = \vee 3, 4, 6, 7.$$

або в мінімальній формі:

$$Z_{t+1} = X_1X_2 + \bar{X}_2Z_t.$$

Таблицю виходів структурного автомата отримують з таблиці переходів абстрактного автомата за таким формальним правилом: на місце стійких станів ставимо відповідний їм вихід автомата, а на місце нестійких станів ставимо риски.

Таблиця виходів структурного автомата, отримана з таблиці 3.2, має вигляд таблиці 3.4.

Ця таблиця є картою Карно рівняння виходу автомата. Тому:

$$Y = \vee 4, 6, 7, (3, 5).$$

або в мінімальній формі

$$Y = Z_t.$$

Таблиця 3.4

		X_1X_2			
		00	01	11	10
Z_t	0	0	0	—	0
	1	1	—	1	1
		Y			

Реалізацію автомата завжди починають з рівнянь переходів. В базисі Шеффера будемо мати схему автомата, наведену на рис. 3.2.

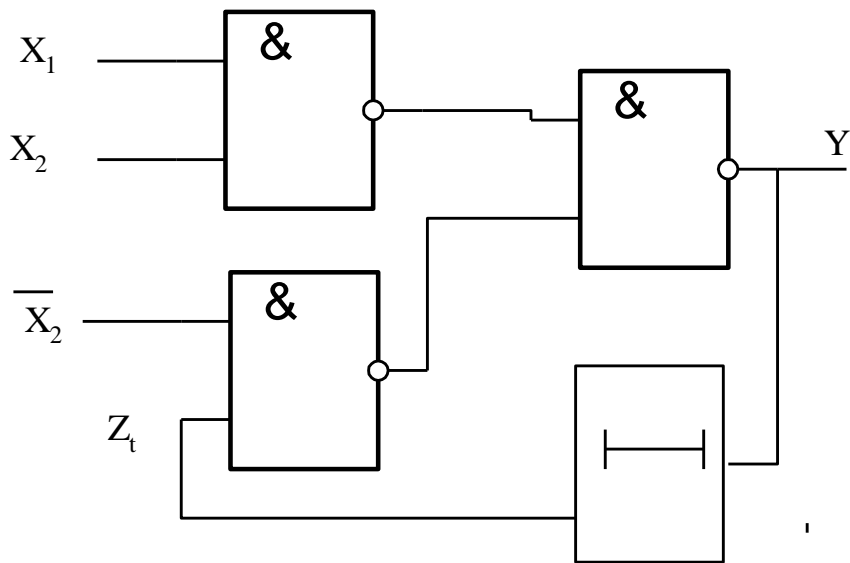


Рис. 3.2

Елемент затримки є зайвим, тому що реальні елементи мають власну затримку на переключення.

Нехай необхідно розробити автомат з одним входом і одним виходом, який працював би за таким алгоритмом: одиниця на вході переводить вихід автомата в протилежний стан; нульовий сигнал на вході зберігає вихід автомата в тому ж стані.

Таблиця переходів абстрактного автомата (табл. 3.5) має три стовпчики та чотири рядки.

Таблиця 3.5

		X		
		0	1	Y
Z ₁	00	[1]	4	0
	01	[2]	3	1
Z ₂	11	1	[3]	0
	10	2	[4]	1

Коли сигнал на вході дорівнює нулю, автомат зберігає на виході попередній стан, який може бути одиничним або нульовим. Тому при нульовому вході будемо мати два стійких стани [1] та [2]. При одиничному вході вихід автомата змінює свій стан на протилежний, який знову ж таки може бути нульовим або одиничним. Тому для одиничного входу відведемо два стійких стана [3] та [4].

Таблиця 3.5 не має еквівалентних станів, тому для реалізації автомата необхідно два елементи пам'яті. Відповідно до цього рядки таблиці переходів будуть мати дворозрядний код. Закодуємо рядки таблиці так, як це прийнято в карті Карно, що полегшує процес мінімізації. Аналіз таблиці показує, що коли автомат знаходиться в стані [4], а вхід змінюється з 1 на 0, то відбувається перехід в стан [2]. При цьому стан автомата змінюється з 10 на 01, тобто переключається одночасно два елементи пам'яті.

Реальні елементи пам'яті мають часову затримку, а тому перехід з 10 на 01 може здійснюватися або за схемою $10 \rightarrow 00 \rightarrow 01$ (перший елемент пам'яті має меншу затримку, ніж другий), або за схемою $10 \rightarrow 11 \rightarrow 01$ (другий елемент пам'яті має меншу затримку, ніж перший). В обох випадках автомат, замість переходу в стан [2], перейде в стан [1], що не відповідає алгоритму його роботи. Таке явище називається гонками. Гонки можуть бути критичні та некритичні. Критичні гонки змінюють алгоритм роботи автомата і є неприпустимими. Некритичні гонки не змінюють алгоритму роботи автомата і на них можна не звертати увагу. Критичні гонки можуть бути лише в тих стовпцях, в яких більше одного стійкого стану.

Усунути критичні гонки можна відповідним кодуванням рядків таблиці переходів. Зміст такого кодування зводиться до того, щоб при переході з одного стану в інший переключався б тільки один елемент пам'яті. При цьому слід мати на увазі, що складність автомата залежить і від способу кодування його станів. Автомат буде мати найбільш оптимальну структуру, якщо рядок, який має найбільшу кількість стійких станів, кодувати нульовою комбінацією. Якщо є декілька таких рядків, то їх вибір довільний.

Оптимальне кодування станів автомата показано в таблиці 3.6.

Таблиця 3.6

		X		
		0	1	Y
Z_{1t}	00	[1]	4	0
	11	[2]	3	1
Z_{2t}	01	1	[3]	0
	10	2	[4]	1

Вона отримана за таким правилом:

- оскільки всі рядки мають однакову кількість сталих станів, кодуємо нульовою комбінацією перший рядок;
- з першого рядка автомат може перейти лише в четвертий. Тому його кодуємо комбінацією, яка на одиницю відрізняється від попередньої, наприклад, комбінацією 10;
- з четвертого рядка автомат може перейти лише в другий, тому код цього рядка буде 11;
- код третього рядка буде тоді 01.

Якщо відповідним кодуванням не вдається усунути гонки, автомат необхідно реалізувати на синхронних елементах пам'яті. Якщо проміжок часу між синхроімпульсами більший за час перехідних процесів в схемі, то гонки не впливають на роботу автомата. Але пам'ять на зворотних зв'язках не може бути синхронною. Крім того, при застосуванні синхронних елементів пам'яті слід дотримуватися викладеної раніше методики кодування, тому що при цьому мінімізується кількість переключень елементів пам'яті, що, в свою чергу оптимізує автомат з точки зору апаратурних витрат.

По таблиці 3.6 побудуємо таблицю переходів структурного автомата (табл. 3.7). В ній рядки 2 та 3 переставлені місцями:

Таблиця 3.7

		X	
		0	1
Z_{1t}	00	00	10
	01	00	01
Z_{2t}	11	11	01
	10	11	10
		$Z_{1(t+1)}$	$Z_{2(t+1)}$

В таблиці перші числа відносяться до першого елемента пам'яті, другі — до другого.

З таблиці маємо рівняння переходів:

$$Z_{1(t+1)} = \bar{X}Z_{1t} + X\bar{Z}_{2t},$$

$$Z_{2(t+1)} = \bar{X}Z_{1t} + XZ_{2t}.$$

Таблиця виходів має вигляд таблиці 3.8.

Таблиця 3.8

		X	
		0	1
Z_{1t}	00	0	—
	01	—	0
Z_{2t}	11	1	—
	10	—	1
		Y	

З цієї таблиці маємо рівняння виходу:

$$Y = Z_{1t}.$$

Схема автомата наведена на рис. 3.3.

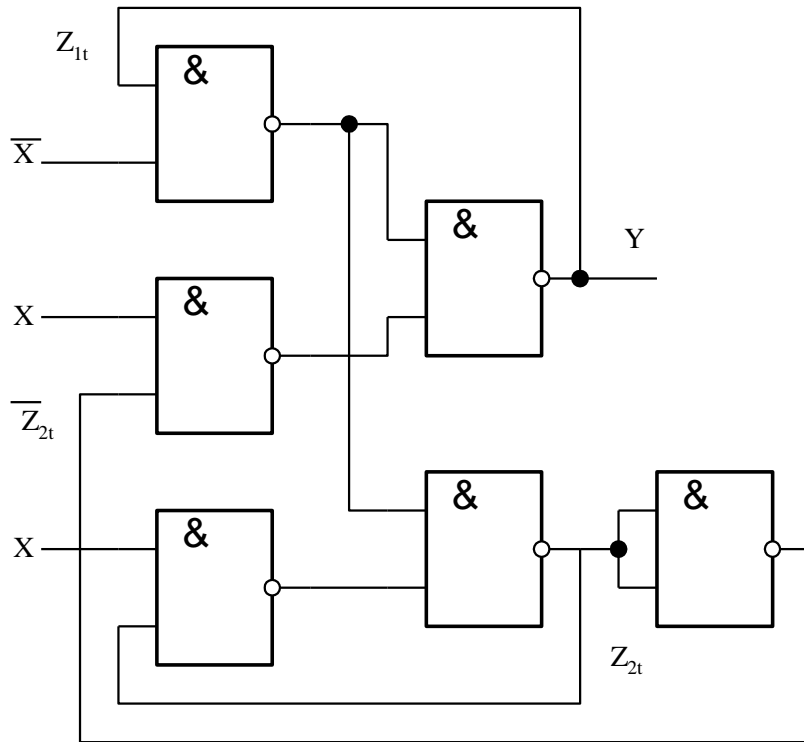


Рис. 3.3

3.2.2. Проектування автоматів на тригерах

Як елементи пам'яті часто застосовують елементарні автомати, які називаються тригерами.

Відомі [15 – 17] типи тригерів, але всі вони можуть бути зведені до 4-х типів: два одноходових та два двохходових.

Одноходовий D -тригер можна описати рівнянням типу:

$$Q_{t+1} = D_t \bar{Q}_t + D_t Q_t = D_t,$$

де Q_{t+1} — наступний стан тригера;

Q_t — стан тригера в даний момент часу;

D_t — вхідний сигнал тригера в даний момент часу.

Для цілей синтезу роботу D -тригера доцільно представити у вигляді таблиці (табл. 3.9).

Таблиця 3.9

$Q_t \rightarrow Q_{t+1}$	D
0 → 0	0
0 → 1	1
1 → 0	0
1 → 1	1

Ця таблиця показує, який сигнал необхідно подати на вхід тригера, для того, щоб його стан змінився з Q_t на Q_{t+1} . В цій таблиці не показано вхід

синхронізації D -тригера (C -вхід). Але слід мати на увазі, що зміна стану тригера відбувається тільки при появі синхроімпульсу на його C -вході.

D -тригер випускається в інтегральному виконанні по 2 та по 4 в одному корпусі.

Одновходовий T -тригер можна представити рівнянням типу:

$$Q_{t+1} = T \cdot \bar{Q}_t + \bar{T} \cdot Q_t.$$

Таблиця, яка зображує це рівняння, має вигляд таблиці 3.10.

T -тригери в інтегральному виконанні не випускаються. При необхідності їх можна отримати з D -тригера, в якому вхід з'єднують з інверсним виходом (рис. 3.4).

Таблиця 3.10

$Q_t \rightarrow Q_{t+1}$	T
$0 \rightarrow 0$	0
$0 \rightarrow 1$	1
$1 \rightarrow 0$	1
$1 \rightarrow 1$	0

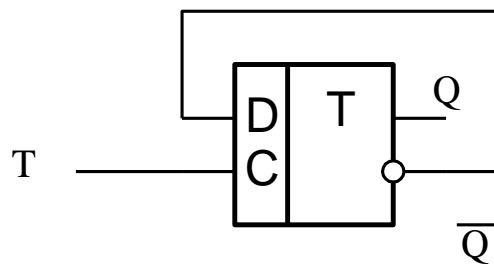


Рис. 3.4

Серед двовходових тригерів найчастіше застосовують RS -тригер. Цей тригер працює за наступним алгоритмом: при $S = 1, R = 0$ тригер встановлюється в одиничний стан; комбінація $S = 0, R = 1$ встановлює тригер в нульовий стан; комбінація $S = 0, R = 0$ зберігає попередній стан тригера; комбінація $S = 1, R = 1$ заборонена, тому що стан тригера стає невизначеним.

Табличне зображення роботи тригера має вигляд таблиці 3.11.

Таблиця 3.11

$Q_t \rightarrow Q_{t+1}$	R	S
$0 \rightarrow 0$	*	0
$0 \rightarrow 1$	0	1
$1 \rightarrow 0$	1	0
$1 \rightarrow 1$	0	*

В таблиці знак “*” означає, що на відповідний вхід можна подавати як нуль, так і одиницю.

Подібні тригери випускаються в інтегральному виконанні. Але є серії мікросхем, в яких такий тригер відсутній. Його можна виконати на елементах “АБО-НІ” (рис. 3.5).

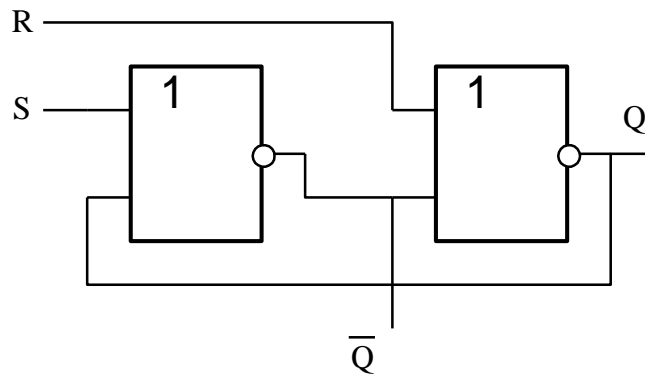


Рис. 3.5

Подібні тригери є тригерами, які управляються одиничними сигналами (елементи з додатною логікою). Існують *RS*-тригери, які управляються нульовими сигналами (мають інверсні входи).

Робота таких тригерів описується наступним алгоритмом: при $S = 1$, $R = 0$ вихід тригера дорівнює одиниці; при $S = 0$, $R = 1$ вихід дорівнює нулю; режим $S = 1$, $R = 1$ є режимом зберігання попереднього стану, а режим $S = 0$, $R = 0$ заборонений. Цей алгоритм зображений таблицею 3.12.

Як такий тригер можна застосувати *D*-тригер з додатковими входами установки в нуль та одиницю (наприклад, тригер К155 ТМ2), зображений на рис. 6.6. В цьому разі входи *D* та *C* не використовуються.

Таблиця 3.12

$Q_t \rightarrow Q_{t+1}$	<i>R</i>	<i>S</i>
$0 \rightarrow 0$	*	1
$0 \rightarrow 1$	1	0
$1 \rightarrow 0$	0	1
$1 \rightarrow 1$	1	*

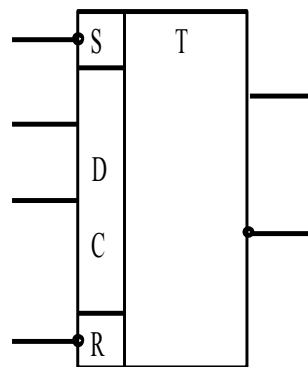


Рис. 3.6

При необхідності тригер з інверсними входами можна виконати на елементах “І-НІ” — елементах Шеффера (рис. 3.7).

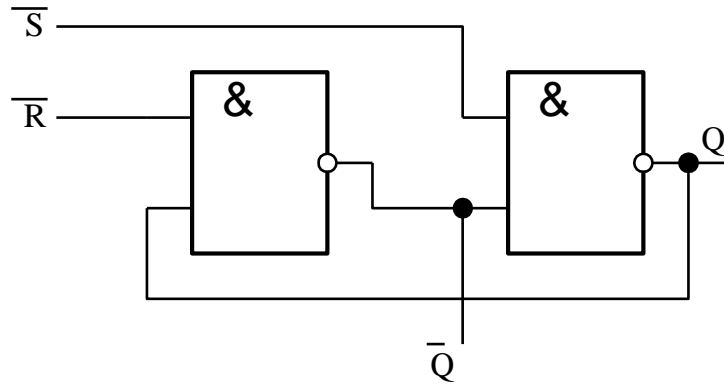


Рис. 3.7

Наступним двовходовим тригером є *JK*-тригер, який називають ще універсальним тригером (рис. 3.8).

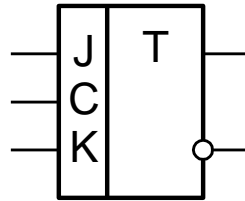


Рис. 3.8

По окремим *J* та *K* входах тригер функціонує так, як *RS*-тригер (*J* — установка в одиницю, *K* — установка в нуль). Якщо входи *J* та *K* об’єднати, то тригер функціонує як *T*-тригер.

Роботу *JK*-тригера можна зобразити таблицею 3.13. *JK*-тригери є синхронними елементами. Випускаються тригери, в яких на входах є схема 3І (наприклад, К155ТВ1) з додатковими входами установки в нуль та одиницю.

Таблиця 3.13

$Q_t \rightarrow Q_{t+1}$	<i>J</i>	<i>K</i>
0 → 0	0	*
0 → 1	1	*
1 → 0	*	1
1 → 1	*	0

Знаки “*” в таблицях, які описують роботу *RS*- та *JK*-тригерів, розглядаються як невикористані набори і дозволяють спростити схему управління тригерами.

Проектування скінчених автоматів на тригерах виконується в тій же послідовності, що й проектування автоматів на зворотних зв’язках.

Таблиця виходів, як і раніше, використовується для отримання рівнянь виходів, вигляд яких не залежить від типу елементів пам'яті.

Таблиця переходів структурного автомата використовується для отримання таблиці збудження. В цій таблиці рядки відповідають станам автомата, а стовпці — вхідним наборам. Елементами таблиці збудження будуть сигнали на входах вибраного типу тригера, якими він переводиться в необхідний стан. Визначення цих сигналів здійснюється таким чином: код рядка таблиці переходів визначає стан тригера Q_t в даний момент часу, а елемент матриці переходів визначає його стан Q_{t+1} в наступний момент часу; в таблиці, яка описує роботу вибраного тригера, вказані сигнали, які необхідно подати на вхід тригера, щоб здійснити його перехід з стану Q_t в стан Q_{t+1} .

Нехай необхідно розробити автомат з одним виходом та двома входами, який виконував би такий алгоритм: якщо спочатку на вході X_2 , а потім на вході X_1 з'явиться одиничний сигнал, то вихідний сигнал теж повинен стати одиничним. Цю задачу можна сформулювати і так: якщо на вході автомата буде спочатку набір 01, а потім — 11, то на виході повинен з'явитися одиничний сигнал.

Таблиця переходів абстрактного автомата зображена таблицею 3.14.

Таблиця 3.14

		$X_1 X_2$				
		00	01	11	10	Y
	3	5	[1]	4	0	
	3	5	[2]	4	1	
	[3]	5	1	4	0	
	3	5	1	[4]	0	
	3	[5]	2	4	0	

Мінімізація станів дозволяє зменшити кількість рядків до трьох. Це вимагає двох елементів пам'яті. Але тоді доцільно мати таблицю з чотирма рядками, один з яких буде заповнено рисками (табл. 3.15).

Таблиця 3.15

		$X_1 X_2$				
		00	01	11	10	Y
	00	[3]	5	[1]	[4]	0
Q_{1t}	11	3	5	[2]	4	1
Q_{2t}	01	3	[5]	2	4	0
	10	-	-	-	-	-

Таблиця виходів не залежить від типу елементів пам'яті і має вигляд таблиці 3.16.

Таблиця 3.16

		X_1X_2			
		00	01	11	10
Q_{1t}	00	0	-	0	0
	01	-	0	-	-
	11	-	-	1	-
	10	-	-	-	-

З цієї таблиці маємо рівняння виходу:

$$Y = \overline{Q}_{1t} \cdot Q_{2t}.$$

Таблиця переходів структурного автомата має вигляд таблиці 3.17.

Таблиця 3.17

		X_1X_2			
		00	01	11	10
Q_{1t}	00	00	01	00	00
	01	00	01	11	00
	11	00	01	11	00
	10	--	--	--	--

$Q_{1(t+1)} \quad Q_{2(t+1)}$

Якщо за елементи пам'яті взяти D -тригер, то таблиця збудження їх буде такою ж, як і таблиця переходів структурного автомата. (Читачеві пропонується перекоонатися в цьому самому). Рівняння управління тригерами мають вигляд:

$$D_1 = X_1X_2Q_{2t};$$

$$D_2 = \overline{X}_1X_2 + X_2Q_{2t} = X_2(\overline{X}_1\overline{Q}_{2t}).$$

Схема автомата наведена на рис. 3.9.

T -тригери, як елементи пам'яті, практично не застосовуються через складність рівнянь і відповідних їм схем.

При застосуванні RS - або JK -тригерів кожному елементу таблиці переходів відповідає два елементи таблиці збудження. Це не зовсім зручно, тому можна рекомендувати складати окремі таблиці для кожного елемента пам'яті.

Таблиця збудження для першого RS -тригера, отримана з таблиць 3.6 та 3.17, представлена таблицею 3.18.

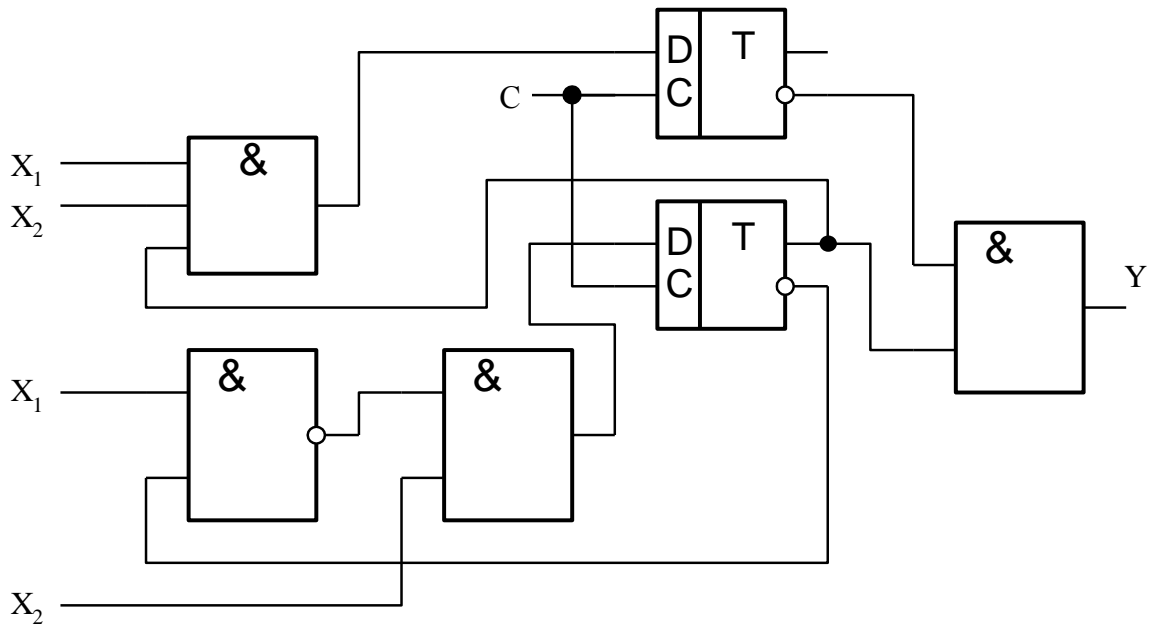


Рис. 3.9

Таблиця 3.18

		X_1X_2			
		00	01	11	10
Q_{1t}	00	*0	*0	*0	*0
	01	*0	*0	01	*0
Q_{2t}	11	10	10	0*	10
	10	--	--	--	--

R_1, S_1

З цієї таблиці маємо рівняння управління першим тригером:

$$R_1 = \bar{X}_1 + \bar{X}_2 = \overline{X_1X_2};$$

$$S_1 = X_1X_2Q_{2t}.$$

Таблиця збудження другого RS-тригера представлена таблицею 3.19.

З цієї таблиці маємо рівняння управління другим тригера:

$$R_2 = \bar{X}_2;$$

$$S_2 = \bar{X}_1X_2.$$

Таблиця 3.19

		X_1X_2			
		00	01	11	10
Q_{1t}	00	*0	01	*0	*0
	01	10	0*	0*	10
Q_{2t}	11	10	0*	0*	10
	10	--	--	--	--

R_2, S_2

Реалізація автомата на *RS*-тригерах наведена на рис. 3.10. Схема автомата менш складна за рахунок розподілу рівнянь включення та виключення тригерів.

Реалізація автомата на *JK*-тригерах виконується за подібною методикою.

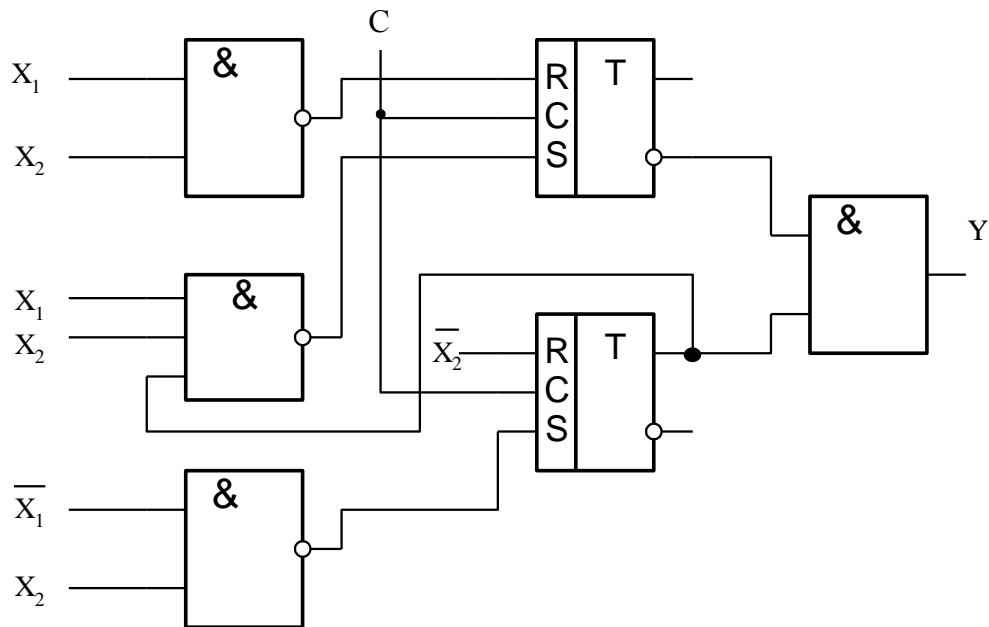


Рис. 3.10

3.3. Проектування автоматів з послідовним входами

3.3.1. Принципи проектування автоматів

Існує досить велика група автоматів, вхідний сигнал яких є послідовний код, часто з невизначеною кількістю розрядів.

Зображення алгоритму роботи таких автоматів у вигляді таблиці переходів недоцільне через їх громіздкість. Більш доцільно алгоритм роботи таких автоматів зображати за допомогою графа, в якому вершини відповідають станам автомата, а дуги — переходам автомата з одного стану в інший. Нестійкі стани на графі не відображаються, тому автомати можна реалізувати лише на синхронних елементах пам'яті.

Граф будь-якого автомата з послідовним входом буде мати вигляд дерева (рис. 3.11), яке утворюється наступним чином: початковий стан автомата b_0 ; якщо в цьому стані на вході автомата з'явиться нульовий сигнал, то він переходить в стан b_1 , якщо одиниця — в стан b_2 ; в цих станах вхідні сигнали переводять автомат або в стан b_3 , або в стан b_4 , або в стан b_5 , або в стан b_6 і т. д.

Реальний автомат повинен мати скінчену кількість станів, тому задача розробника полягає в тому, щоб зробити граф закінченим. Це можливо, якщо вихідні дуги останнього ярусу завести на попередні вершини. Ця задача має два аспекти: визначення кількості ярусів та визначення вершин, на які необхідно завести вихідні дуги останнього ярусу.

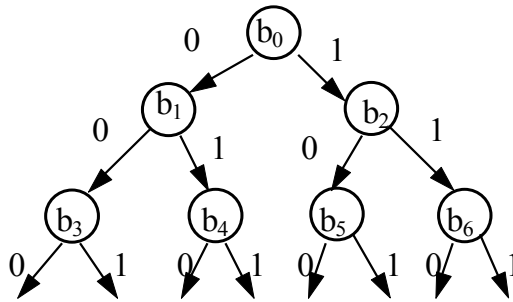


Рис. 3.11

Кількість ярусів часто можна визначити за алгоритмом роботи автомата, якщо в ньому визначена кількість розрядів, яка підлягає обробці. Якщо алгоритм роботи автомата таких вказівок не має, то кількість ярусів визначається наступним чином: беруть їх мінімальну кількість (два яруси) і перевіряють можливість виконання заданого алгоритму роботи; якщо це не вдається, беруть ще один ярус, і т.д.

Вирішити, яку дугу і на яку вершину необхідно завести, в загальному випадку неможливо. Це треба вирішувати для кожного конкретного автомата окремо.

Нехай необхідно розробити автомат з одним входом і одним виходом. На вхід автомата надходить нескінченна і довільна послідовність нулів та одиниць. На виході автомата повинен з'явитися одиничний сигнал, якщо десятковий еквівалент чотирьох розрядів дорівнює або більше 10, причому, на кожному наступному такті перший розряд виштовхується, а замість четвертого з'являється новий. Схема обробки сигналів показана на рис. 3.12.



Рис. 3.12

Як видно з рисунка, на такті j оброблюються три розряди з такту $j-1$.

Граф роботи автомата зображений на рис. 3.13. На кожній дузі проставляємо значення входу (чисельник) та виходу (знаменник). Стани, на які заводяться вихідні дуги, визначені наступним чином:

- нульова дуга стану b_7 відповідає випадку, коли на вхід автомата поступила послідовність 0000; на наступному такті буде послідовність 000 X (X або одиниця, або нуль); три розряди 000 цієї послідовності відповідають стану b_7 ; тому цю дугу заведемо на b_7 ;

- одинична дуга стану b_7 відповідає випадку, коли на вхід автомата надійшла послідовність 0001; на наступному такті буде послідовність 001 X ; три розряди 001 цієї послідовності відповідають стану b_8 ; тому цю дугу заведемо на стан b_8 .

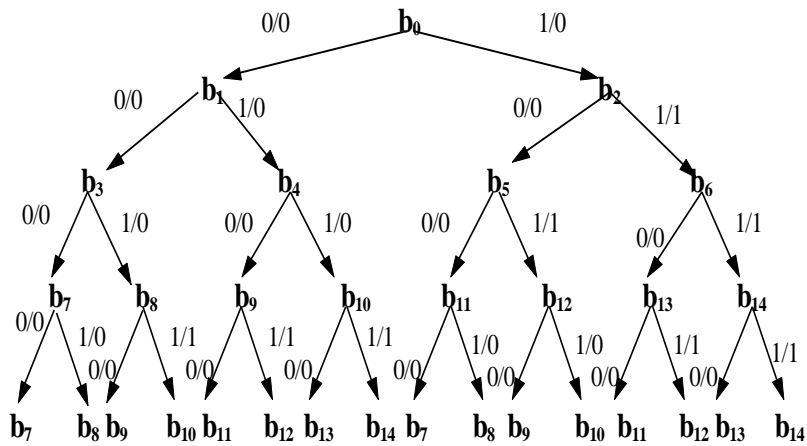
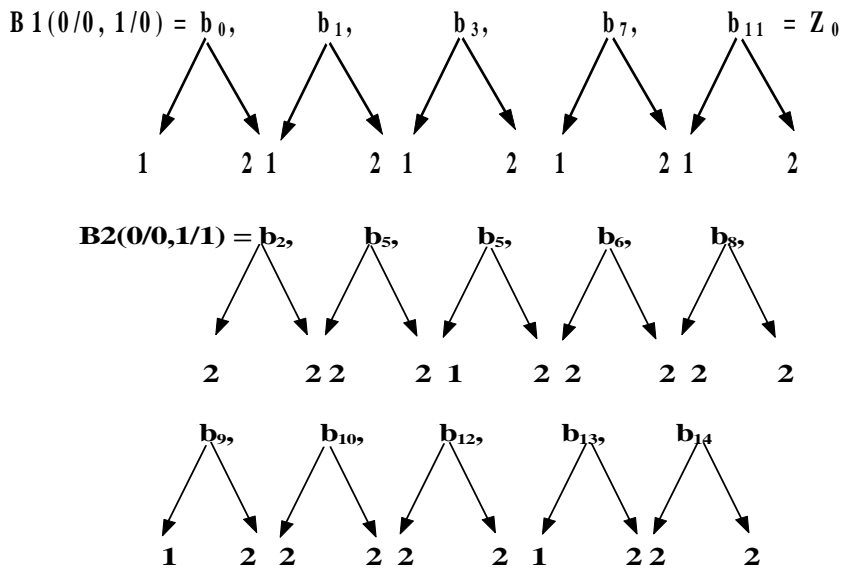


Рис. 3.13

Стани, на які замикається решта дуг, визначені аналогічно. Маємо автомат з 15-ма станами, які визначають необхідність 4-х елементів пам'яті. Для зменшення їх кількості необхідно провести мінімізацію станів автомата. Для автомата з послідовними входами цю процедуру необхідно проводити класичним методом, який полягає в тому, що вся множина станів розподіляється на не еквівалентні класи. Стани кожного класу перевіряються на еквівалентність. В початковому стані маємо два класи не еквівалентних вершин:

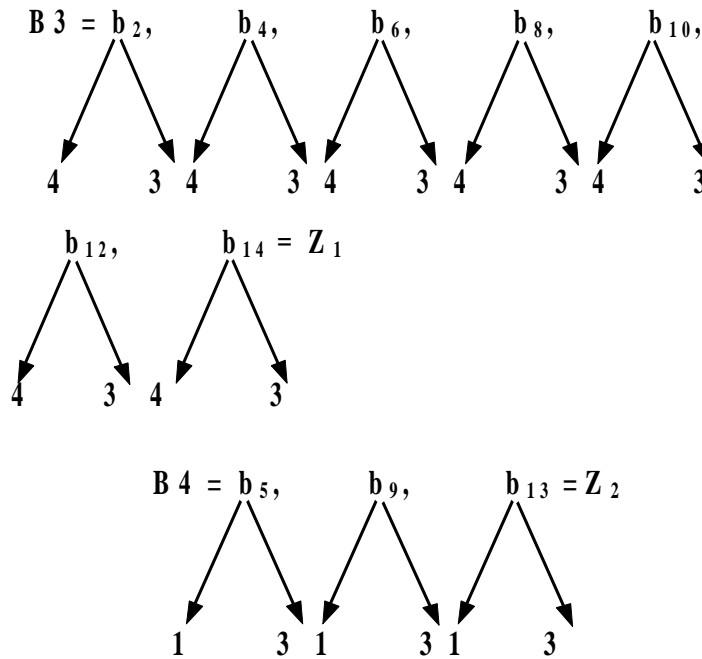


Ці класи не еквівалентні тому, що при одиничному входному сигналі у них різні значення виходу.

Для перевірки еквівалентності вершин одного класу необхідно при даному стані подати на вхід автомата нульовий (ліва стрілка) та одиничний (права стрілка) сигнали і визначити, в який клас переходить цей стан. Наприклад, якщо в стані b_0 на вхід автомата подати нульовий сигнал, то він переходить в стан b_1 , який належить до першого класу (до множини B_1).

Тому на кінці лівої стрілки ставимо цифру 1. Якщо в цьому ж стані на вхід автомата подати одиничний сигнал, то він переходить в стан b_2 , який належить до другого класу (до множини $B2$). Тому на кінці правої стрілки ставимо цифру 2.

Такий підхід показує, що вершини множини $B1$ еквівалентні і їх можна об'єднати в одну вершину Z_0 . Вершини множини $B2$ не еквівалентні. Розіб'ємо цю множину на явно не еквівалентні частини:



Вершини множини $B3$ еквівалентні і їх можна об'єднати в одну вершину Z_1 .

Вершини множини $B4$ теж еквівалентні і їх можна об'єднати в одну вершину Z_2 . Таким чином 15 станів автомата зведені до трьох, що вимагає лише двох елементів пам'яті.

Тепер є можливість побудувати таблицю переходів абстрактного автомата та таблицю його виходів (табл. 3.20), в якій в чисельнику проставлено стан автомата в наступний момент часу, а в знаменнику — вихід автомата.

Таблиця 3.20

X

	0	1
Z_0	$Z_0/0$	$Z_1/0$
Z_1	$Z_2/0$	$Z_1/1$
Z_2	$Z_0/0$	$Z_1/1$

Ця таблиця заповнюється таким чином:

- візьмемо будь-яку вершину з Z_0 , наприклад, b_0 ;
- подамо при цьому стані на вхід автомата нуль;
- автомат переходить в стан b_1 , який належить до Z_0 і має нульовий вихід;
- у відповідній клітині таблиці проставимо $Z_0/0$;
- подамо на вхід автомата одиницю;
- стан b_0 переходить в стан b_2 , який належить до Z_1 і має вихід 0;
- у відповідній клітині таблиці ставимо $Z_1/0$.

Решта клітин таблиці заповнюється аналогічно. Закодуємо стани автомата так (правила кодування будуть розглянуті далі):

$$Z_1 = 00;$$

$$Z_2 = 01;$$

$$Z_0 = 11.$$

Тоді таблиця переходів структурного автомата має вигляд (табл. 3.21).

Таблиця 3.21

		X	
		0	1
Q_{1t}	00	01	00
	01	11	00
Q_{2t}	11	11	00
	10	- -	- -

Якщо елементи пам'яті реалізувати на D -тригерах, то таблиця 3.21 буде одночасно і таблицею збудження цих тригерів. Тоді рівняння управління тригерами мають вигляд:

$$D_1 = \bar{X}Q_{2t};$$

$$D_2 = \bar{X}.$$

Таблиця виходів має вигляд табл. 3.22.

Таблиця 3.22

		X	
		0	1
Q_{1t}	00	0	1
	01	0	1
Q_{2t}	11	0	0
	10	-	-

З цієї таблиці отримуємо рівняння виходу:

$$Y = X\bar{Q}_{1t}.$$

Схема автомата зображена на рис. 3.14.

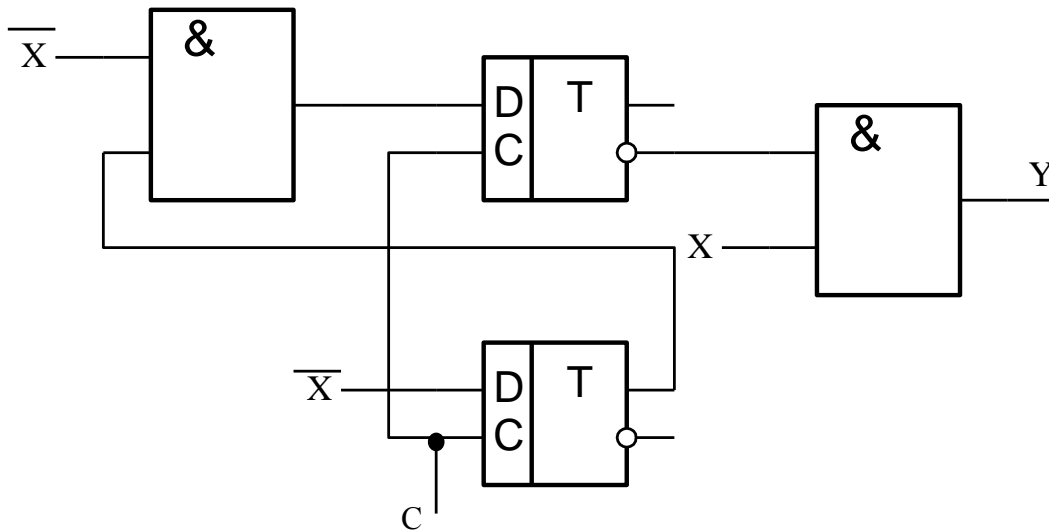


Рис. 3.14

3.3.2. Оптимальне кодування станів автоматів

Відсутність гонок в автоматах з послідовним входом дозволяє застосувати метод кодування його станів, який забезпечує мінімальну кількість елементів для його реалізації. Відомо [7, 18 – 20], що складність автомата визначається кількістю переключень елементів пам'яті. Тому кодування повинно бути таким, щоб забезпечити мінімум переключень елементів пам'яті.

Алгоритм оптимального кодування зводиться до наступного:

1. Будуємо граф, вершини якого відповідають станам автомата, а дуги — переходам з одного стану в інший.
2. Кожній дузі ставимо у відповідність вагу, яка чисельно дорівнює кількості переходів з одного стану в інший.

3. Вводимо функцію $W = \sum_{i=1}^n p(i, j) \cdot d(i, j)$,

де $d(i, j)$ — відстань Хемінга між кодами вершин;

$p(i, j)$ — вага дуги між вершинами i та j .

4. Будуємо матрицю T :

$$T = \begin{vmatrix} \alpha_1 & \beta_1 \\ \alpha_R & \beta_R \end{vmatrix}$$

яка складається з пар вершин, з'єднаних дугами, тобто пар станів, між якими в автоматі є переходи.

5. Впорядковуємо рядки матриці T за таким правилом:

- в перший рядок поміщаємо пару $\alpha_i \beta_i$ з найбільшою вагою; якщо таких пар декілька, то вибираємо з них пару з найбільшою сумою частот

(частотою називається кількість повторення даної вершини в матриці T); якщо і таких пар декілька, вибір серед них довільний;

- в другий рядок поміщуємо пару з максимальною вагою, але при умові, що вона має спільну вершину з першою парою;

- наступні рядки формуються за тим же правилом, але пари повинні мати спільний елемент з будь-якою парою в попередніх рядках.

6. Вершини першої пари кодуються кодами $K_{\alpha 1} = 00\dots 0$, $K_{\beta 1} = 00\dots 1$, тобто числами 0 та 1. Нульову комбінацію коду буде мати вершина з максимальною частотою.

7. Викреслюємо перший рядок матриці. Отримаємо M' матрицю, в якій один елемент вже закодовано.

8. Побудуємо матрицю M_j , вибираючи з M' пари, які мають елемент, що потребує кодування, та вже закодовані елементи.

9. Візьмемо коди K_j з тих, які ще не використовувалися і для кожного з них визначимо функцію W . Для кодування елемента виберемо код, для якого ця функція мінімальна.

10. Продовжуємо цю процедуру, доки не будуть закодовані всі елементи.

Нехай маємо автомат, граф якого після мінімізації станів зображений на рис. 3.15.

Побудуємо матрицю T , в якій вкажемо вагу пар та суму частот її елементів. Внутрішні дуги не враховуються. Частота елементів:

$$Z_1 = 1;$$

$$Z_2 = 3;$$

$$Z_3 = 3;$$

$$Z_4 = 3;$$

$$Z_5 = 3;$$

$$Z_6 = 3.$$

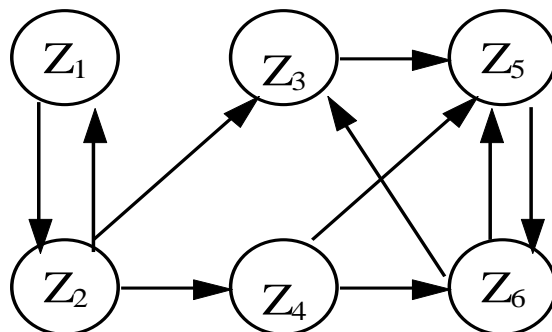


Рис. 3.15

$$T = \begin{array}{|c|c|c|c|} \hline Z_1 & Z_2 & 2 & 4 \\ \hline Z_2 & Z_3 & 1 & 6 \\ \hline Z_2 & Z_4 & 1 & 6 \\ \hline Z_3 & Z_5 & 1 & 6 \\ \hline Z_3 & Z_6 & 1 & 6 \\ \hline Z_4 & Z_5 & 1 & 6 \\ \hline Z_4 & Z_6 & 1 & 6 \\ \hline Z_5 & Z_6 & 2 & 6 \\ \hline \end{array}$$

Впорядкуємо елементи і побудуємо матрицю M :

$$M = \begin{array}{|c|c|c|} \hline Z_5 & Z_6 & 2 \\ \hline Z_4 & Z_6 & 1 \\ \hline Z_4 & Z_5 & 1 \\ \hline Z_3 & Z_6 & 1 \\ \hline Z_3 & Z_5 & 1 \\ \hline Z_2 & Z_3 & 1 \\ \hline Z_1 & Z_2 & 2 \\ \hline Z_2 & Z_4 & 1 \\ \hline \end{array}$$

Закодуємо

$$K_5 = 000,$$

$$K_6 = 001.$$

В другому рядку є незакодований елемент Z_4 . Побудуємо матрицю M_4 :

$$M_4 = \begin{array}{|c|c|} \hline Z_4 & Z_6 \\ \hline Z_4 & Z_5 \\ \hline \end{array}$$

Вільні коди:

$$C_2 = 010,$$

$$C_5 = 101,$$

$$C_3 = 011,$$

$$C_6 = 110,$$

$$C_4 = 100,$$

$$C_7 = 111.$$

Якщо $K_4 = 010$, то $W_2 = 2 \cdot 1 + 1 \cdot 1 = 3$.

Якщо $K_4 = 011$ $W_3 = 3$.

Якщо $K_4 = 100$ $W_4 = 3$.

Якщо $K_4 = 101$ $W_5 = 4$.

Якщо $K_4 = 110$ $W_6 = 8$.

Якщо $K_4 = 111$ $W_7 = 7$.

Кодами елемента Z_4 можуть бути 011, 100 та 010. Приймаємо $K_4 = 010$. Третій рядок має незакодований елемент Z_3 . Його матриця має вигляд:

$$M_3 = \begin{vmatrix} Z_3 & Z_6 \\ Z_3 & Z_5 \end{vmatrix}$$

Вільні коди: C_3, C_4, C_5, C_6 та C_7 . Для них маємо:

$$W_3 = 2,$$

$$W_4 = 3,$$

$$W_5 = 3,$$

$$W_6 = 5,$$

$$W_7 = 5.$$

Вибираємо

$$K_3 = 011.$$

Наступним незакодованим елементом буде елемент Z_2 . Його матриця має вигляд:

$$M_2 = \begin{vmatrix} Z_2 & Z_3 \\ Z_2 & Z_4 \end{vmatrix}$$

Для них

$$W_4 = 5,$$

$$W_5 = 5,$$

$$W_6 = 3,$$

$$W_7 = 3.$$

Приймаємо $K_2 = 110$.

Маємо лише один незакодований елемент Z_1 . Його матриця має вигляд:

$$M_1 = \begin{vmatrix} Z_1 & Z_2 \end{vmatrix}$$

$$W_4 = 2,$$

$$W_5 = 4,$$

$$W_7 = 4.$$

Приймаємо $K_1 = 100$.

Остаточно маємо:

$$Z_1 = 100,$$

$$Z_2 = 110,$$

$$Z_3 = 011,$$

$$Z_4 = 010,$$

$$Z_5 = 000$$

$$Z_6 = 001$$

3.3.3. Проектування багатовходових автоматів

У загальному випадку автомати з послідовними входами можуть мати декілька входів, на кожний з яких надходить безперервна послідовність нулів та одиниць. Граф, який відображає роботу такого автомата, буде мати вершини, з яких виходить 2^m ребер (m — кількість входів автомата). Синтезувати такий автомат дуже важко, тому класичний підхід зводиться до

декомпозиції (розподілу) автомата на частини, кожна з яких має один вхід. Виходи підавтоматів об'єднуються ще одним підавтоматом, який є автоматом з паралельними входами. При цьому алгоритм роботи кожного з підавтоматів повинен бути таким, щоб не порушити алгоритм роботи автомата в цілому.

Алгоритму розподілу або декомпозиції автомата на підавтомати не існує, тому цю задачу необхідно вирішувати для кожного конкретного випадку на основі досвіду та інтуїції.

Нехай необхідно розробити автомат з двома входами та одним виходом. На входи X_1 та X_2 надходять довільні та незалежні один від одного послідовності нулів та одиниць. На виході автомата повинен з'явитися одиничний сигнал, якщо спочатку на вході X_2 , а потім на вході X_1 , з'явиться парна одиниця.

Розіб'ємо автомат на три підавтомата (рис. 3.16).

Підавтомати A_1 та A_2 є автомати з послідовними входами і повинні видавати одиницю на своїх виходах після появи парної одиниці на входах. Підавтомат A_3 є автоматом з паралельними входами і повинен видавати одиницю на виході тоді, коли, спочатку на вході Y_1 , а потім на вході Y_2 , з'явиться одиниця.

Розглянемо роботу автомата A_1 . Одиниця на його виході повинна з'явитися після кожної парної одиниці на вході, незалежно від того, будуть між ними нулі чи ні, тобто коли на вході є сполучення 11, 101, 1001 і т. д. Якщо після парної одиниці на вході буде нуль, то вихід автомата повинен зберігати одиницю. Це пояснюється тим, що він повинен дати змогу автомату A_2 оцінити ситуацію і видати відповідний сигнал.

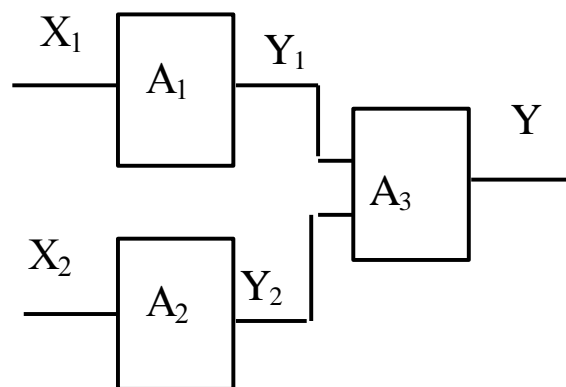


Рис. 3.16

Автомат A_1 є автомат з послідовним входом і його роботу можна зобразити у вигляді графа. Оскільки невідома кількість розрядів, що обробляється, візьмемо мінімальну кількість вершин (рис. 3.17).

Щоб зробити граф скінченим, замкнемо його вихідні дуги на попередні вершини.

Дуга 1, яка відповідає послідовності одних лише нулів, можна завести на вершини b_1 або на b_2 (заведемо на b_2).
 Дуги 2 та 3, які відповідають одній одиниці на вході, можна завести лише на вершину b_3 . Дуга 4, яка відповідає двом одиницям на вході, завести, без порушення алгоритму роботи автомата, нікуди, тому що немає вершини з дугою 0/1. Тому ця кількість вершин недостатня для реалізації заданого алгоритму роботи автомата.

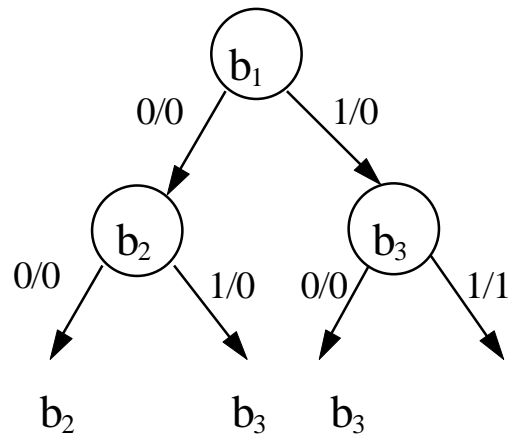


Рис. 3.17

Збільшимо кількість вершин графа (рис. 3.18).

Алгоритм роботи автомата виконується, тому 7 станів достатньо для його реалізації.

Граф має три типи дуг, тому його стани розіб'ємо на три не еквівалентні класи:

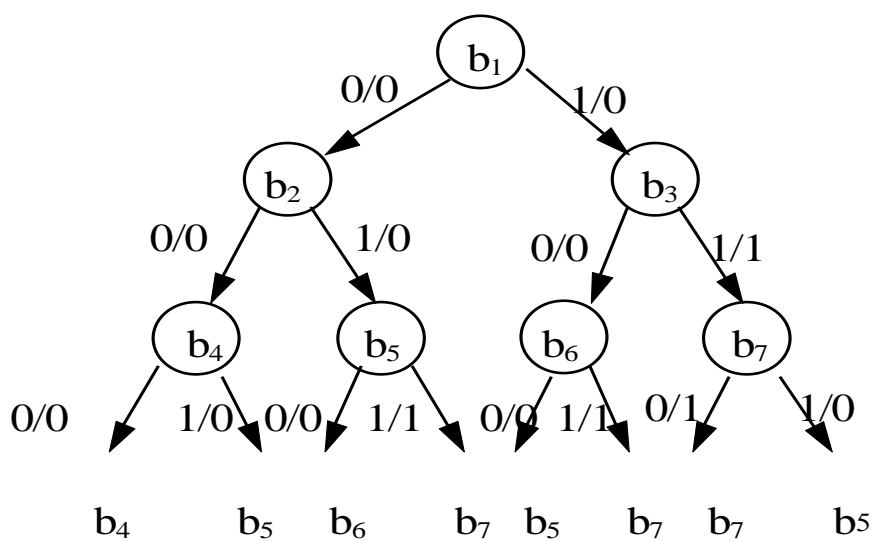


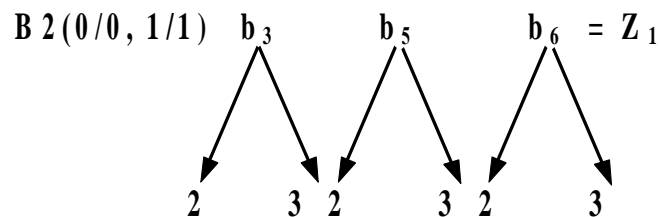
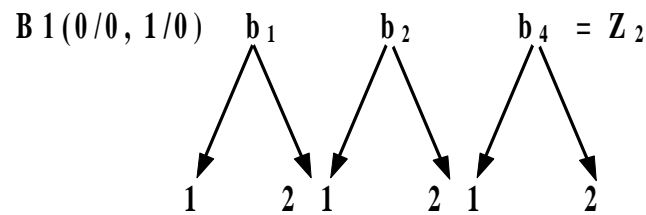
Рис. 3.18

$$B1(0/0, 1/0) = b_1, b_2, b_4;$$

$$B2(0/0, 1/1) = b_3, b_5, b_6$$

$$B3(0/1, 1/0) = b_7 = Z_2$$

Перевіримо еквівалентність вершин множини $B1$.



Для реалізації автомата необхідно два елементи пам'яті. Складемо таблицю переходів та виходів абстрактного автомата (табл. 3.23).

Таблиця 3.23

	X_1	
	0	1
Z_0	$Z_1/0$	$Z_2/0$
Z_1	$Z_2/0$	$Z_3/1$
Z_2	$Z_3/1$	$Z_2/0$

Методом оптимального кодування визначимо, що:

$$Z_1 = 11, \quad Z_2 = 00, \quad Z_3 = 01.$$

Взявши як елементи пам'яті D -тригер, отримаємо таблицю збудження їх (табл. 3.24).

Таблиця 3.24

	X_1	
	0	1
Q_{1t}	00	01
Q_{2t}	01	00
	11	00
	--	--

З цієї таблиці маємо рівняння управління тригерами:

$$D_1 = \bar{X}_1 Q_{1t};$$

$$D_2 = \bar{X}_1 Q_{2t} + X Q_{2t} = X \oplus Q_{2t}.$$

Таблиця виходів має вигляд таблиці 3.25.

Таблиця 3.25

		X_1	
		0	1
Q_{1t}	00	0	1
	01	1	0
Q_{2t}	11	0	0
	10	-	-

Тоді рівняння виходу має вигляд:

$$Y = \overline{Q_{1t}}(X \oplus Q_{2t}).$$

Схема автомата $A1$ наведена на рис. 3.18.

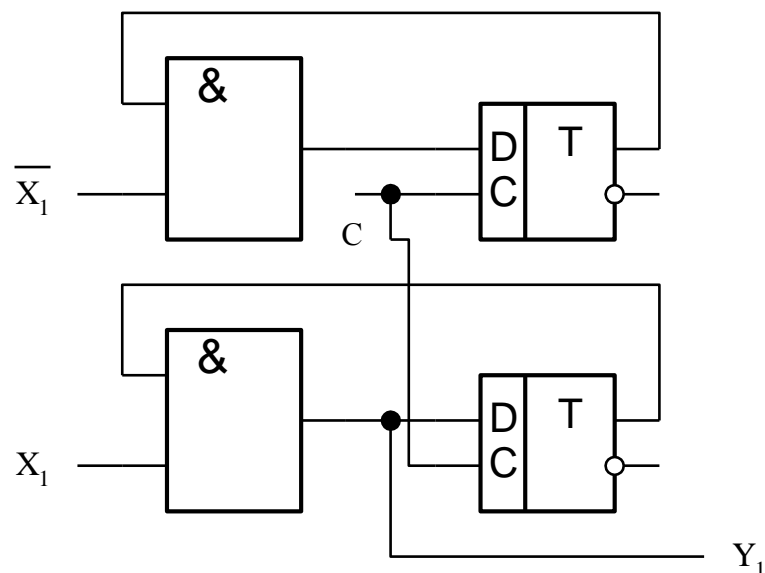


Рис. 3.18

Автомат $A2$ відрізняється від $A1$ тим, що його вихід не повинен зберігати одиницю на виході, коли після парної одиниці на вході з'явиться нуль. Граф автомата зображений на рис. 3.19.

Нульову дугу з вершини b_2 можна завести на b_1 або на b_2 . Заведемо її на b_2 . Одиничну дугу b_2 можна завести тільки на b_3 . Якщо нульову дугу b_3 завести на b_3 , а одиличну – на b_2 , то автомат буде виконувати заданий алгоритм.

Тому цієї кількості вершин досить для реалізації заданого алгоритму роботи автомата.

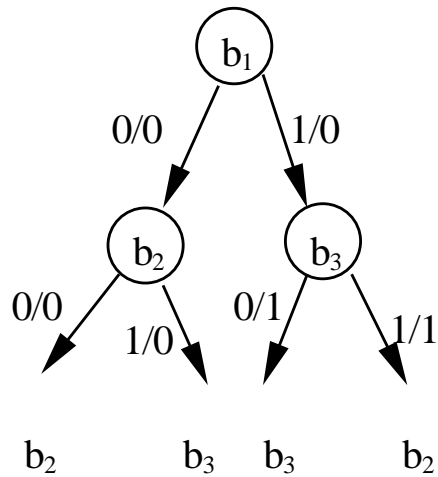
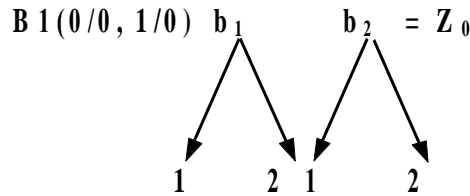


Рис. 3.19

Маємо два класи вершин:



$$B2(0/0, 1/1) = b_3 = Z_1$$

Таблиця переходів та виходів абстрактного автомата має вигляд таблиці 3.26.

Таблиця 3.26

		X_2	
		0	1
Z_0	Z_0	$Z_0/0$	$Z_1/0$
	Z_1	$Z_1/0$	$Z_0/1$

Коди станів:

$$Z_0 = 0,$$

$$Z_1 = 1$$

При застосуванні як елемента пам'яті D -тригера таблиця збудження його має вигляд таблиці 3.27.

Таблиця 3.27

		X_2	
		0	1
Q_t	0	0	1
	1	0	1

Рівняння управління тригером має вигляд:

$$D = X_2 \oplus Q_t.$$

Таблиця виходу автомата має вигляд таблиці 3.28.

Таблиця 3.28

		X_2	
		0	1
Q_t	0	0	0
	1	0	1

Рівняння виходу має вигляд:

$$Y_2 = X_2 Q_t.$$

Схема автомата наведена на рис. 3.20.

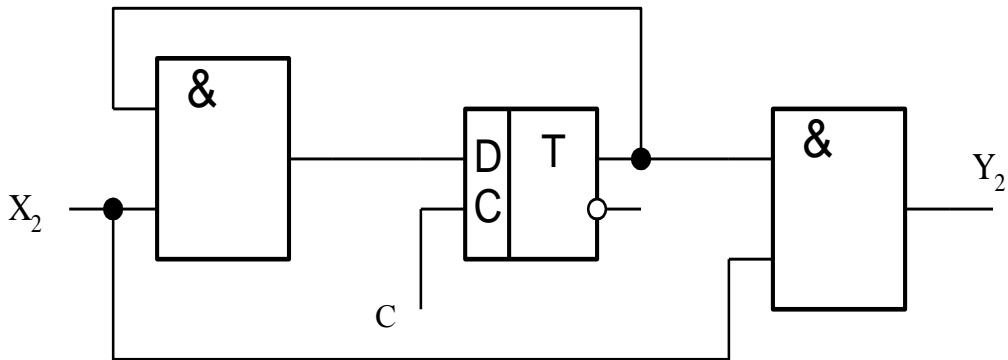


Рис. 3.20

Синтез автомата АЗ був наведений в підрозділі 3.2.

3.4. Контрольні питання до розділу 3

1. Дайте визначення скінченного автомату.
2. Охарактеризуйте методика проектування скінчених автоматів на зворотніх зв'язках.
3. Які типи тригерів можуть бути використані для побудови автоматів з паралельними входами?
4. Що таке таблиця переходів скінчених автоматів?
5. Назвіть основні принципи проектування скінчених автоматів з послідовними входами.
6. Дайте характеристику графів роботи скінчених автоматів з послідовними входами.
7. Як закодувати всі стани скінчених автоматів?
8. Особливості проектування багатовходових скінчених автоматів.
9. Чим відрізняються стійкі та нестійкі стани скінчених автоматів?

РОЗДІЛ 4 ПРОЕКТУВАННЯ МІКРОПРОГРАМНИХ АВТОМАТІВ

4.1. Загальні відомості про мікропрограмні автомати

Комбінаційні схеми та скінчені автомати — це апаратурна реалізація заданого алгоритму роботи системи управління. Цей алгоритм визначає типи елементів та зв'язки між ними і не може бути змінений в процесі роботи.

В сучасних системах управління основним є застосування програмних методів, які дозволяють створювати системи, здатні при одній і тій же апаратурній реалізації вирішувати широке коло задач. Такі системи мають програмний блок, в якому зберігається необхідна програма і яку можна замінити, не змінюючи інші частини системи.

Виконання алгоритму роботи в подібних системах здійснюється послідовно в часі. Кожний крок в цій послідовності називається мікрооперацією або оператором. Подібні системи називаються мікропрограмними автоматами (МПА), причому слово “мікро” визначає обмежену кількість операторів — від декількох сотень до декількох тисяч.

МПА, крім програмного блоку, має функціональні (ФБ) та логічні (ЛБ) блоки. Перші призначені для виконання команд управління виконавчими механізмами, а другі здійснюють перевірку стану об'єкта управління (перевірку стану відповідних датчиків) або перевірку виконання тих чи інших операцій.

Розробка структури МПА починається з розробки програми його роботи в такій послідовності:

1. Визначення виконавчих механізмів об'єкта управління. Команди включення або виключення цих механізмів є виходами МПА. В системах управління технологічними процесами або робототехнічними комплексами (РТК) такими виконавчими механізмами будуть контактори (реле) електродвигунів, електро-, гідро- або пневмоклапани, регулюючі клапани, захвати РТК тощо. В системах обробки інформації виконавчими механізмами будуть арифметико-логічні пристрої, регістри, дешифратори, мультиплексори, елементи запам'ятовуючих пристроїв тощо. Ці елементи мають управляючі входи, за допомогою яких здійснюється той чи інший вид обробки інформації. Команди управління таких елементів є багаторозрядні двійкові числа. Кожна комбінація таких чисел є певною командою (зсув інформації вліво або вправо, зберігання, установка в нуль або в одиницю тощо), яку можна позначити одним символом.

2. Визначення параметрів, які характеризують хід технологічного процесу (або процесу обробки інформації), а також визначення величини сигналів датчиків, при яких виконується зміна положення регулюючих органів. Ці датчики формують вхідні сигнали МПА і визначають склад логічних блоків.

3. Розробка програми роботи МПА, яка визначає послідовність включення або виключення виконавчих механізмів в залежності від стану

датчиків. Для забезпечення розробки такої програми кожний функціональний та логічний оператор позначається певним символом.

4. Налагодження програми роботи МПА. Мета цього налагодження – зменшення кількості кроків програми або кількості мікрооперацій. В процесі складання програми досить складно уникнути повторювання мікрооперацій, тому що головна задача цього етапу – правильно описати послідовність виконання кроків програми, враховуючи всі можливі ситуації, які можуть виникнути на об'єкті. Теорія МПА дозволяє спростити програму і зменшити час на її виконання.

5. Розробка структурної схеми МПА. Ця стадія проектування носить формальний характер, тому що структура автомата визначається його програмою.

6. Розробка принципової схеми МПА. Перехід від структурної до принципової схеми визначається вибором відповідної елементної бази. Правильний вибір елементів забезпечує їх мінімальну кількість, що дозволяє зменшити габаритні розміри системи управління, зменшити потужність споживання та збільшити надійність автомата.

4.2. Форми представлення алгоритму роботи мікропрограмних автоматів

Мікропрограмний автомат можна розглядати як скінчений автомат і задавати алгоритм його роботи за допомогою таблиць переходів та виходів. Але при великій кількості змінних це недоцільно через велику розмірність таких таблиць. Тому розроблені інші форми представлення алгоритму роботи МПА.

Однією з таких форм є завдання алгоритму роботи МПА у вигляді графа, в якому вершини відповідають операторам, а дуги визначають послідовність їх виконання. Така форма представлення алгоритму роботи МПА називається граф-схемою алгоритмів (ГСА) [18].

Будь-який процес управління має початок і закінчення, тому необхідно виділити одну початкову і одну вихідну вершини. Початкова вершина не має вхідних дуг, вихідні – вихідних дуг. Решта вершин відповідає функціональним та логічним операторам. Функціональна вершина має лише одну вихідну дугу. Логічна вершина має дві вихідні дуги: нульову, яка відповідає невиконаній логічній умові, та одиничну, яка відповідає виконаній логічній умові. Функціональні вершини позначаються великими літерами (будь-якого алфавіту) або однією буквою з індексом (A_1, A_2) Логічні вершини позначаються малими літерами або однією літерою з індексом. Функціональні оператори є команди включення (виключення) функціональних блоків і відповідають вихідним сигналам Z_{xi} .

Як функціональні, так і логічні оператори можуть виконувати дії над різними частинами інформації.

Нехай маємо процес управління, зображений на рис. 4.1.

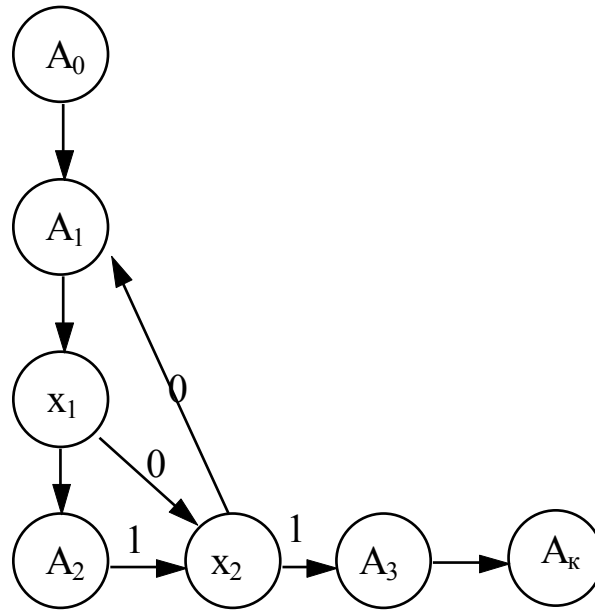


Рис. 4.1

В залежності від стану датчиків X_1 та X_2 послідовність команд управління буде наступною:

- $X_1 = 0, X_2 = 0$ $A_0 \rightarrow A_1 \rightarrow \dots \rightarrow A_1$
- $X_1 = 1, X_2 = 0$ $A_0 \rightarrow A_1 \rightarrow A_2 \rightarrow A_1 \rightarrow A_2 \dots$
- $X_1 = 0, X_2 = 1$ $A_0 \rightarrow A_1 \rightarrow A_3 \rightarrow A_k$
- $X_1 = 1, X_2 = 1$ $A_0 \rightarrow A_1 \rightarrow A_2 \rightarrow A_3 \rightarrow A_k$

Граф-схема алгоритмів може бути початковою інформацією для синтезу МПА. Така методика викладена в [3]. При великій кількості операторів ГСА втрачає наочність, що ускладнює і процес синтезу. Тому була запропонована форма запису програми роботи МПА [14], при якій оператори випиуються в рядок зліва направо в порядку їх виконання. Праворуч логічного оператора випиуюється той оператор, який виконується тоді, коли значення логічного дорівнює одиниці. Праворуч логічного оператора проставляється стрілка, кінець якої направлений вверх і який закінчується номером. Ліворуч оператора, до якого йде ця стрілка, ставиться стрілка, кінець якої направлений вниз і який має той же номер. Нумерація стрілок довільна. Може бути декілька стрілок, направлених вверх, з одним і тим же номером, але стрілка, з тим же номером і направлена вниз, може бути лише одна. Стрілка є не що інше, як розірвана дуга на ГСА, кінці розриву якої мають один і той же номер. Подібна форма запису програми роботи МПА називається логічною схемою алгоритмів (ЛСА) і досить широко застосовується.

ЛСА по рис. 4.1. буде мати вигляд:

$$U = A_0 \downarrow_2 A_1 x_1 \uparrow^1 A_2 \downarrow_1 x_2 \uparrow^2 A_3 A_k.$$

ЛСА є певна форма запису програми роботи МПА. Щоб скласти таку програму, необхідно провести змістовний опис операторів, тобто позначити відповідними буквами всі оператори.

Позначення та кодування операторів виконує сам розробник, що дозволяє розробляти МПА і не спеціалісту в області програмування.

Не всі процеси управління можна записати за допомогою лише функціональних, логічних операторів та стрілок. Нехай маємо ГСА, зображену на рис. 4.2.

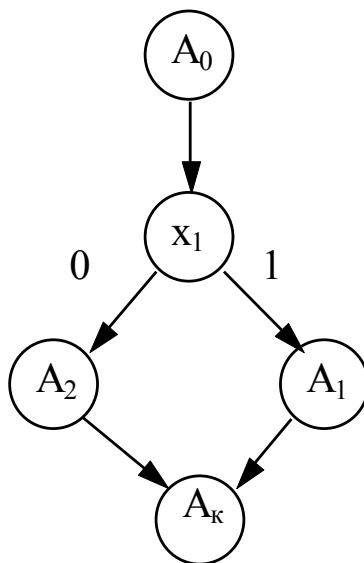


Рис. 4.2

Спробуємо записати цей процес управління у вигляді ЛСА:

$$U = A_0 x_1 \uparrow^1 A_1 A_k.$$

В цій ЛСА немає куди поставити оператор A_2 і нікуди повести стрілку 1. Спробуємо іншу форму:

$$U = A_0 \bar{x}_1 \uparrow^1 A_2 A_k.$$

Тепер немає місця для оператора A_1 .

В звичайному програмуванні для переходу з декількох місць програми в одну її точку застосовують оператор безумовного переходу (GOTO n). Такий оператор є необхідним і в ЛСА. Він позначається наступним чином — $\omega \uparrow^n$. Це перехід по стрілці без перевірки умов переходу. Тоді процес управління по рис. 4.2 має вигляд:

$$U = A_0 x_1 \uparrow^1 A_1 \omega \uparrow^2 \downarrow_1 A_2 \downarrow_2 A_k. \quad (4.1)$$

Послідовність виконання функціональних операторів в ЛСА визначається значенням логічних операторів, які розглядаються як двійкові змінні і які можуть приймати значення нуля або одиниці. Вони можуть входити в ЛСА як в прямому так і в інверсному виді. Так, (4.1) може бути записана у вигляді:

$$U = A_0 \bar{x}_1 \uparrow^1 A_2 \omega \uparrow^2 \downarrow_1 A_1 \downarrow_2 A_k.$$

У загальному випадку логічні оператори можуть бути булевою функцією деякої кількості змінних.

4.3. Структурна схема мікропрограмних автоматів

Найбільш відомою та зручною структурною схемою МПА є структурна схема автомата Уілкса [10, 18 – 20] (рис. 4.3).

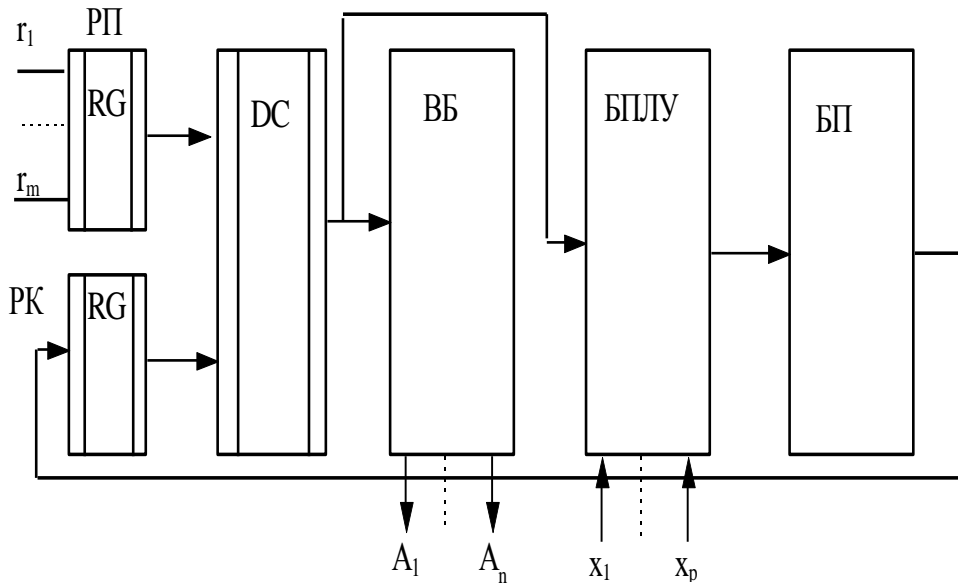


Рис. 4.3

Схема автомата складається з регістра програм РП, регістра команд РК, дешифратора ДС, вихідного блоку ВБ, блоку перевірки логічних умов БПЛУ та блоку пам'яті БП.

В регістрі РП зберігається код програми, яка виконується в даний момент часу. Його можна змінювати за допомогою змінних $r_1 \div r_m$.

В регістрі РК зберігається код команди, яка виконується в даний момент часу. Ця команда розшифровується дешифратором ДС.

Блок ВБ є вихідним блоком МПА. Якщо вихідні команди $A_1 \div A_n$ (функціональні оператори) необхідно запам'ятати, блок ВБ буде мати відповідну кількість тригерів. Якщо команду не потрібно запам'ятовувати, тригери відсутні.

Блок БПЛУ визначає стан логічних операторів і є вхідним блоком МПА. Для перевірки стану логічних операторів можна застосувати досить просту схему (рис. 4.4).

При відсутності команди на перевірку $Z_p = 0$ на виході обох елементів "Г" сигнал дорівнює нулю. При $Z_p = 1$ (перевірити) одиничний сигнал з'явиться на виході одного з елементів, в залежності від значення сигналу логічного оператора (стану датчика).

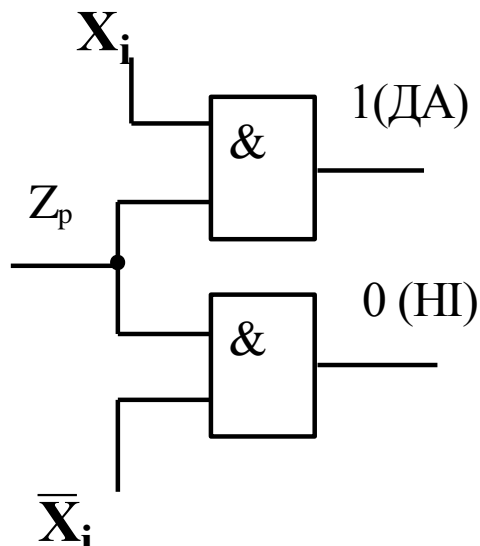


Рис. 4.4

Програмний блок БП зберігає коди всіх команд усіх програм.

Автомат Уїлкса працює наступним чином: розшифровується код команди, яка виконується в даний момент часу. В залежності від стану логічних умов з БП заноситься в регістр РК код нової команди, яка буде виконуватися і т. д.

Серед входів БП можуть бути входи кнопок “ПУСК” та “СТОП”, за допомогою яких автомат можна включати та виключати.

Схема автомата Уїлкса тісно пов’язана з ЛСА. Кількість виходів дешифратора дорівнює кількості операторів в цій ЛСА. Кількість виходів $A_1 \div A_n$ дорівнює кількості функціональних операторів. Кожному логічному операторові відповідає схема перевірки за рис. 7.4. Якщо ЛСА має N елементів, то розрядність коду команди дорівнює $\log_2 N$ і цьому ж числу відповідає розрядність регістра РК. Об’єм пам’яті БП дорівнює $N \cdot \log_2 N$.

Таким чином, ЛСА визначає витрати апаратури на реалізацію МПА. Тому основна задача розробника зводиться до отримання ЛСА з мінімальною кількістю операторів.

4.4. Складання логічної схеми алгоритму

ЛСА процесу управління складають на основі словесного опису алгоритму роботи автомата. На цьому етапі отримати ЛСА з мінімальною кількістю операторів досить важко. Тому головну увагу необхідно звернути на отримання ЛСА, яка б правильно описувала процес управління. При цьому необхідно враховувати, що функціональні оператори можуть бути двох типів. Якщо функціональний оператор виконується за один такт (наприклад, зсув числа в регістрі, запис числа в регістр тощо), то запам’ятовувати цю команду не потрібно. В протилежному випадку оператор слід запам’ятати за допомогою тригера. Але в цьому разі повинна бути

команда на виключення тригера, що вимагає додаткового оператора, який можна позначити як A_i^* . Подібна ситуація часто зустрічається при управлінні механізмами технологічних процесів.

При складанні початкової ЛСА можна рекомендувати всі функціональні оператори запам'ятовувати. Далі буде показано, як можна вилучати з ЛСА зайві з них.

ЛСА складають послідовним записом всіх команд та перевірок у відповідності з заданим алгоритмом. Процес складання ЛСА розглянемо на прикладі.

Нехай необхідно скласти ЛСА процесу управління виготовленням бетонної плити. Схема технологічної ділянки наведена на рис. 4.5.

Виготовлення бетонної плити зводиться до заповнення дерев'яної форми (опалубки) бетонною сумішшю, яка надходить з бункерів Б1 ÷ Б3. Коли бункер Б2 знаходиться над отворами плити, видача бетону з нього припиняється. Коли бункери знаходяться над виступом, подача бетону припиняється, а форма опускається вниз. Після заповнення виступу форма знову піднімається.

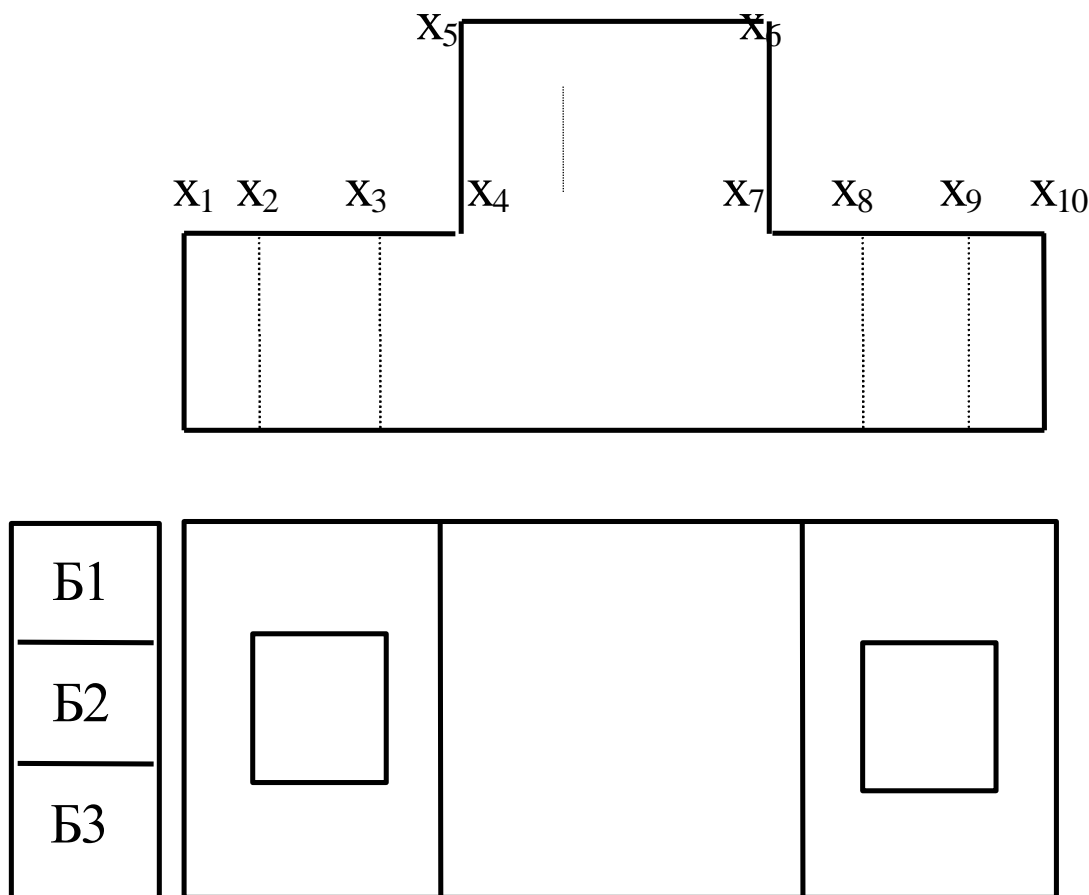


Рис. 4.5

Коли бункери знаходяться над відповідним місцем форми, переключаються датчики $x_1 \div x_{10}$. Повне заповнення форми відбувається за два рази — форма спочатку рухається ліворуч, а потім праворуч.

Складемо змістовний опис операторів:

A_1 — включення двигуна руху ліворуч;

A_2 — включення двигуна руху праворуч;

A_3 — включення двигуна руху вниз;

A_4 — включення двигуна руху вверх;

A_5, A_6, A_7 — відкриття затворів бункерів Б1 ÷ Б3;

A_i^* — виключення i -ї команди.

При $x_i = 1$ бункери знаходяться над відповідним місцем плити.

В початковому положенні $x_1 = 1$. Для зручності розіб'ємо процес на дві частини: рух плити ліворуч та рух її праворуч.

ЛСА управління рухом плити ліворуч має вигляд:

$$\begin{aligned}
 U_{\text{Л}} = & \downarrow_1 A_0 x_1 \uparrow^1 A_1 A_5 A_7 \downarrow_2 A_6 x_2 \uparrow^2 \downarrow_3 A_6^* x_3 \uparrow^3 \downarrow_4 A_6 \\
 & x_4 \uparrow^4 A_1^* A_5^* A_7^* A_6^* \downarrow_5 A_3 x_5 \uparrow^5 A_3^* \downarrow_6 A_1 A_5 A_7 A_6 x_6 \uparrow^6 A_1^* A_5^* A_7^* \\
 & A_6^* \downarrow_7 A_4 x_7 \uparrow^7 A_4^* \downarrow_8 A_1 A_5 A_7 A_6 x_8 \uparrow^8 \downarrow_9 A_6^* x_9 \uparrow^9 \downarrow_{10} A_6 x_{10} \uparrow^{10} A_1^* A_5^* A_7^* A_6^* A_{k1}.
 \end{aligned} \quad (4.2)$$

Оператор A_{k1} є умовним оператором закінчення першої частини процесу заповнення плити бетоном.

ЛСА другої частини процесу управління має вигляд:

$$\begin{aligned}
 U_{\text{П}} = & \downarrow_1 A_{k1} x_{10} \uparrow^1 A_2 A_5 A_7 \downarrow_2 A_6 x_9 \uparrow^2 \downarrow_3 A_6^* x_8 \uparrow^3 \\
 & \downarrow_4 A_6 x_7 \uparrow^4 A_2^* A_5^* A_7^* A_6^* \downarrow_5 A_3 x_6 \uparrow^5 A_3^* \downarrow_6 A_2 A_5 A_7 A_6 \\
 & x_5 \uparrow^6 A_2^* A_5^* A_7^* A_6^* \downarrow_7 A_4 x_4 \uparrow^7 A_4^* \downarrow_8 A_2 A_5 A_7 A_6 x_3 \uparrow^8 \\
 & \downarrow_9 A_6^* x_2 \uparrow^9 \downarrow_{10} A_6 x_1 \uparrow^{10} A_2^* A_5^* A_7^* A_6^* A_k.
 \end{aligned} \quad (4.3)$$

В даному технологічному процесі лише один датчик може бути в одиничному стані. Це дозволяє запропонувати програму роботи автомата, при якій він перевіряє стани датчиків і видає необхідні команди:

$$\begin{aligned}
 U'_{\text{Л}} = & A_0 \downarrow_1 \bar{x}_1 \uparrow^2 \bar{x}_2 \uparrow^3 \bar{x}_3 \uparrow^4 \bar{x}_4 \uparrow^5 \bar{x}_5 \uparrow^6 \bar{x}_6 \uparrow^7 \\
 & \bar{x}_7 \uparrow^9 \bar{x}_8 \uparrow^3 \bar{x}_9 \uparrow^4 \bar{x}_{10} \uparrow^{10} \omega \uparrow^1 \downarrow_2 A_1 A_5 A_7 \downarrow_4 A_6 \omega \uparrow^1 \\
 & \downarrow_3 A_6^* \omega \uparrow^1 \downarrow_5 A_3 \downarrow_8 A_1^* A_5^* A_7^* \omega \uparrow^3 \downarrow_6 A_3^* \omega \uparrow^2 \downarrow_7 A_4 \omega \uparrow^8 \\
 & \downarrow_9 A_4^* \omega \uparrow^2 \downarrow_{10} A_1^* A_5^* A_7^* A_6^* A_{k1}.
 \end{aligned} \quad (4.4)$$

ЛСА (4.2) та (4.3) мають по 42 оператори (оператори A_0 та $\omega \uparrow^n$ не враховуються). ЛСА (4.4) має 26 операторів. Але робити остаточні висновки про доцільність тієї чи іншої форми ЛСА рано. Для цього необхідно провести спрощення ЛСА.

В технологічних лініях часто застосовують так звані крокові механізми. Такі механізми після подачі команди на включення здійснюють переміщення на заданий крок (обертаються на заданий кут) і зупиняються під дією датчика

кроку. Особливістю таких механізмів є те, що початок і кінець їх руху відбувається при одному і тому ж стані датчика (наприклад, при $X = 1$). Тому звичайний опис управління ним на мові ЛСА типу

$$\downarrow_n A_i x \uparrow^n A_i^*.$$

для них не підходить. При подачі команди A_i на включення механізм буде негайно виключено, тому що $X = 1$. Для таких механізмів необхідна перевірка початку руху, після чого можна перевіряти кінець руху. Тоді ЛСА управління таким механізмом має вигляд:

$$\downarrow_{n1} A_i \bar{x} \uparrow^{n1} \downarrow_{n2} x \uparrow^{n2} A_i^*.$$

4.5. Спрощення логічної схеми алгоритму

Кількість членів ЛСА визначає апаратні витрати на реалізацію МПА. Тому необхідно добитися, щоб їх кількість була мінімальною, але без порушення алгоритму роботи автомата.

Спрощення ЛСА проводиться в три етапи. На першому етапі ті функціональні оператори, які в усій ЛСА виконуються разом, можна замінити на один оператор. Це впливає з можливості управління декількома механізмами однією командою.

В ЛСА (4.2 ÷ 4.4) можна замінити:

$$A_5 A_7 = A_8,$$

$$A_5^* A_7^* = A_8^*.$$

Об'єднати ці оператори з A_6 неможливо, тому що останній зустрічається в ЛСА без перших. Включення в цю групу оператора A_1 було б можливо, якби (4.2) була б єдиною програмою управління — в ЛСА (4.3) замість оператора A_1 стоїть оператор A_2 .

На другому етапі спрощення ЛСА здійснюється за рахунок так званих “очікуючих” операторів. Очікуючі оператори — це оператори типу:

$$\downarrow_n A_i x_p \uparrow^n A_j, \tag{4.5}$$

$$\downarrow_n A_i x_p \uparrow^n A_i^*. \tag{4.6}$$

Особливість цих операторів полягає в тому, що МПА очікує, доки параметр x_p буде мати певне значення (в даному випадку доки x_p не стане одиницею). Такі оператори відображають рефлекторний зв'язок (зв'язок по певному значенню контрольованого параметра) між окремими операціями або між початком і кінцем однієї і тої ж операції.

Оператори (4.5) спростити неможливо. Але в операторах (4.6) можна скоротити A_i^* при умові, що A_i не буде запам'ятовуватися.

Замість змінних x_p в (4.5) та (4.6) може стояти часова затримка:

$$\downarrow_n A_i x_t \uparrow^n A_j, \tag{4.7}$$

$$\downarrow_n A_i x_t \uparrow^n A_i^*. \tag{4.8}$$

Такі пари відображають часовий зв'язок між окремими операціями або між початком і кінцем однієї і тої ж операції. В (4.8) оператор A_i^* теж можна скоротити при умові, що A_{\square} не буде запам'ятовуватися.

Часові затримки створюються таймерами (реле часу). Для їх включення необхідна команда, що необхідно враховувати при складанні ЛСА.

В ЛСА (4.2) є дві очікуючі пари: $\downarrow_5 A_3 x_5 \uparrow^5 A_3^*$ та $\downarrow_7 A_4 x_7 \uparrow^7 A_4^*$ з яких можна виключити A_3^* та A_4^* . В ЛСА (4.3) є пари $\downarrow_5 A_3 x_6 \uparrow^5 A_3^*$ та $\downarrow_7 A_4 x_4 \uparrow^7 A_4^*$ з яких також можна виключити ці оператори.

З урахуванням першого та другого етапів ЛСА (4.2) та (4.3) мають вигляд:

$$\begin{aligned}
 U_{II} = & \downarrow_1 A_0 x_1 \uparrow^1 A_1 A_8 \downarrow_2 A_6 x_2 \uparrow^2 \downarrow_3 A_6^* x_3 \uparrow^3 \downarrow_4 A_6 \\
 & x_4 \uparrow^4 A_1^* A_8^* A_6^* \downarrow_5 A_3 x_5 \uparrow^5 \downarrow_6 A_1 A_8 A_6 x_6 \uparrow^6 A_1^* A_8^* A_6^* \\
 & \downarrow_7 A_4 x_7 \uparrow^7 \downarrow_8 A_1 A_8 A_6 x_8 \uparrow^8 \downarrow_9 A_6^* x_9 \uparrow^9 \downarrow_{10} A_6 x_{10} \uparrow^{10} \\
 & A_1^* A_8^* A_6^* A_{k1}.
 \end{aligned} \tag{4.9}$$

$$\begin{aligned}
 U_{II} = & \downarrow_1 A_{k1} x_{10} \uparrow^1 A_2 A_8 \downarrow_2 A_6 x_9 \uparrow^2 \downarrow_3 A_6^* x_8 \uparrow^3 \downarrow_4 A_6 \\
 & x_7 \uparrow^4 A_2^* A_8^* A_6^* \downarrow_5 A_3 x_6 \uparrow^5 \downarrow_6 A_2 A_8 A_6 x_5 \uparrow^6 A_2^* A_8^* A_6^* \downarrow_7 A_4 \\
 & x_4 \uparrow^7 \downarrow_8 A_2 A_8 A_6 x_3 \uparrow^8 \downarrow_9 A_6^* x_2 \uparrow^9 \downarrow_{10} A_6 x_1 \uparrow^{10} A_2^* A_8^* A_6^* A_k.
 \end{aligned} \tag{4.10}$$

$$\begin{aligned}
 U'_{II} = & A_0 \downarrow_1 \bar{x}_1 \uparrow^2 \bar{x}_2 \uparrow^3 \bar{x}_3 \uparrow^4 \bar{x}_4 \uparrow^5 \bar{x}_5 \uparrow^6 \bar{x}_6 \uparrow^7 \bar{x}_7 \uparrow^9 \\
 & \bar{x}_8 \uparrow^3 \bar{x}_9 \uparrow^4 \bar{x}_{10} \uparrow^{10} \omega \uparrow^1 \downarrow_2 A_1 A_8 \downarrow_4 A_6 \omega \uparrow^1 \downarrow_3 A_6^* \omega \uparrow^1 \\
 & \downarrow_5 A_3 \downarrow_8 A_1^* A_8^* \omega \uparrow^3 \downarrow_6 A_3^* \omega \uparrow^2 \downarrow_7 A_4 \omega \uparrow^8 \downarrow_9 A_4^* \omega \uparrow^2 \\
 & \downarrow_{10} A_1^* A_8^* A_6^* A_{k1}.
 \end{aligned} \tag{4.11}$$

В ЛСА (4.4) ці оператори скоротити неможливо, тому що МПА не чекає зміни логічних операторів, а весь час перевіряє їх стан по черзі.

На третьому етапі спрощення ЛСА здійснюється за рахунок усунення повторюваності операторів. Це вимагає представлення ЛСА у вигляді так званих формул переходів.

Формула переходів будь-якого оператора A_i є формула типу:

$$A_i \rightarrow \bigvee_{k=1}^n a_k A_j, \tag{4.12}$$

де a_k – набір логічних умов, при яких після оператора A_i виконується оператор A_j .

Множина формул переходів для всіх функціональних операторів ЛСА утворює систему формул переходів і є ще однією формою завдання алгоритму роботи МПА. Формули переходів отримують з ЛСА.

Кожна формула переходів має наступні властивості:

$$\bigvee_{k=1}^n a_{ik} = 1, \quad (4.13)$$

$$a_{ik}a_{jk} = 0. \quad (4.14)$$

Умова (4.13) говорить про те, що після оператора A_i виконується не більше одного оператора при заданих логічних умовах. Умова (4.14) говорить про те, що після A_i завжди виконується якийсь оператор.

Нехай маємо ЛСА:

$$U = A_0 \downarrow_4 \bar{x}_1 \uparrow^1 x_2 \uparrow^2 A_1 A_3 \bar{x}_1 \uparrow^1 \bar{x}_2 \uparrow^3 \downarrow_2 A_2 \downarrow_3 A_1 A_3 \bar{x}_1 \uparrow^1 \bar{x}_2 \uparrow^3 A_2 \downarrow_3 A_1 A_3 x_1 \uparrow^4 \downarrow_1 A_k. \quad (4.15)$$

Її формули переходів мають вигляд:

$$\begin{aligned} A_0 &\rightarrow \downarrow_1 (\bar{x}_1 x_2 A_1 + \bar{x}_1 x_2 A_2 + x_1 A_k); \\ A_1 &\rightarrow A_3; \\ A_2 &\rightarrow A_1; \\ A_3 &\rightarrow (\bar{x}_1 \bar{x}_2 A_2 + \bar{x}_1 x_2 A_1 + x_1 A_k) = \omega \uparrow^1. \end{aligned} \quad (4.16)$$

В системі (4.16) формули переходів для операторів A_0 та A_3 однакові. Одну з них замінимо на оператор $\omega \uparrow^n$, стрілка від якого йде до першого оператора. Якщо однакові вирази є в декількох формулах переходів, то всі вони, крім одного, замінюються на оператор безумовного переходу.

Перехід від формул переходів до ЛСА здійснюється за наступним правилом:

- * виписуємо оператор A_0 ;
- * з його формули переходів беремо будь-який її член (крім того, який закінчується оператором A_k), який закінчується оператором A_j ;
- * звертаємося до формули переходів оператора A_j і виписуємо з його формули переходів будь-який його член;
- * ці операції повторюємо доки не будуть виписані всі оператори ЛСА;
- * якщо необхідно виписати оператор, який уже виписаний раніше, робимо до нього перехід за допомогою оператора безумовного переходу.

Для системи (4.16) маємо:

$$U = A_0 \downarrow_1 \bar{x}_1 \uparrow^1 x_2 \uparrow^1 A_1,$$

звертаємося до формули переходів оператора A_1 :

$$U = A_0 \downarrow_1 \bar{x}_1 \uparrow^1 x_2 \uparrow^1 A_1 A_3 \omega \uparrow^1.$$

Нові оператори виписати неможливо, тому звертаємося до формули переходів A_0 і по ній визначаємо, які оператори будуть виконуватися при переході по стрілках при змінних x_1 та x_2 . Визначаємо, що при $\bar{x}_1 \bar{x}_2$ виконується оператор A_2 :

$$U = A_0 \downarrow_1 \bar{x}_1 \uparrow^1 x_2 \uparrow^2 A_1 A_3 \omega \uparrow^1 \downarrow_2 A_2.$$

Після оператора A_2 виконується оператор A_1 , який уже входить в ЛСА. Тому зробимо перехід до нього за допомогою оператора безумовного переходу:

$$U = A_0 \downarrow_1 \bar{x}_1 \uparrow x_2 \uparrow^2 \downarrow_3 A_1 A_3 \omega \uparrow^1 \downarrow_2 A_2 \omega \uparrow^3.$$

При $x_1 = 1$ виконується A_k . Остаточно маємо ЛСА:

$$U = A_0 \downarrow_1 \bar{x}_1 \uparrow^4 x_2 \uparrow^2 \downarrow_3 A_1 A_3 \omega \uparrow^1 \downarrow_2 A_2 \omega \uparrow^3 \downarrow_4 A_k.$$

Ця ЛСА має 5 операторів замість 15 в (4.16).

Випадок, коли окремі формули переходів однакові, зустрічається досить рідко. Класична теорія МПА рекомендує скорочувати ЛСА за рахунок елементарних дужок, які утворюються при винесенні за дужки однакових змінних. Елементарна дужка є вираз типу:

$$A_i \rightarrow x_k (x_p A_j + \bar{x}_p A_m).$$

Однакові елементарні дужки в усіх формулах переходів (крім одної) замінюються оператором безумовного переходу. Це можливо, якщо усі члени всіх формул переходів будуть мати однакові змінні, чого можна добитися розширенням функцій. Розширення функцій — операція зворотна мінімізації — досягається перемноженням кожної змінної на вираз $(x_j + \bar{x}_j)$. При великій кількості змінних це досить складна операція, а отримані формули переходів будуть дуже громіздкими. Можна запропонувати більш простий алгоритм спрощення ЛСА за формулами переходів, який пояснимо на прикладі ЛСА (4.9).

Переведемо ЛСА (4.9) в формули переходів. При цьому будемо мати на увазі, що в даному технологічному процесі може бути в одиничному стані лише один датчик (або ні одного). Якщо всі датчики знаходяться в нульовому стані, то виконується той оператор, який виконувався останнім. Якщо в системі можуть одночасно переключатися декілька датчиків, то при записі формул переходів необхідно мати на увазі умови (4.13) та (4.14).

$$A_0 \rightarrow x_1 A_1 + \bar{x}_1 A_0,$$

$$A_1 \rightarrow A_8,$$

$$A_8 \rightarrow A_6,$$

$$A_6 \rightarrow x_2 A_6^* + x_4 A_1^* + x_6 A_1^* + x_8 A_6^* + x_{10} A_1^*,$$

$$A_6^* \rightarrow x_3 A_6 + x_4 A_3 + x_6 A_4 + x_9 A_6 + x_{10} A_{k1},$$

$$A_1^* \rightarrow A_8^*,$$

$$A_8^* \rightarrow A_6^*,$$

$$A_3 \rightarrow x_5 A_1 + \bar{x}_5 A_3,$$

$$A_4 \rightarrow x_7 A_1 + \bar{x}_7 A_4.$$

Перейдемо від цієї системи формул переходів знову до ЛСА:

$$U_{\%o'} = A_0 \uparrow_1 A_1 A_8 A_6.$$

В формулі переходів оператора A_6 є 5 змінних. Операція розширення приведе до утворення з них 32 наборів типу:

$$\tilde{x}_2 \tilde{x}_4 \tilde{x}_6 \tilde{x}_8 \tilde{x}_{10},$$

де знак “~” означає, що змінна може бути в прямому або інверсному стані. З 32-х наборів реальними можуть бути лише 6 з них:

набори	номер набору
$\bar{x}_2 \bar{x}_4 \bar{x}_6 \bar{x}_8 \bar{x}_{10} A_6$	(0)
$x_2 \bar{x}_4 \bar{x}_6 \bar{x}_8 \bar{x}_{10} A_6^*$	(16)
$\bar{x}_2 x_4 \bar{x}_6 \bar{x}_8 \bar{x}_{10} A_1^*$	(8)
$\bar{x}_2 \bar{x}_4 x_6 \bar{x}_8 \bar{x}_{10} A_1^*$	(4)
$\bar{x}_2 \bar{x}_4 \bar{x}_6 x_8 \bar{x}_{10} A_6^*$	(2)
$\bar{x}_2 \bar{x}_4 \bar{x}_6 \bar{x}_8 x_{10} A_1^*$	(1)

З цих 6 наборів і буде складатися формула переходів оператора A_6 . В ЛСА необхідно ввести один з них. Візьмемо нульовий набір.

$$U_{\mathcal{L}} = A_0 x_1 \uparrow A_1 A_8 \downarrow_1 A_6 \bar{x}_2 \uparrow \bar{x}_4 \uparrow \bar{x}_6 \uparrow \bar{x}_8 \uparrow \bar{x}_{10} \uparrow \omega \uparrow^1.$$

Виписати новий оператор неможливо, тому звертаємося до формули переходів оператора A_6 і нумеруємо стрілки. Якщо x_2 буде мати значення одиниці, після A_6 виконується оператор A_6^* . Звертаємося до формули переходів цього оператора і застосовуємо до нього ту ж саму процедуру, що й до оператора A_6 . Остаточна ЛСА буде мати вигляд:

$$\begin{aligned} U_{\mathcal{L}} = & \downarrow_2 A_0 x_1 \uparrow^2 A_1 A_8 \downarrow_1 A_6 \bar{x}_2 \uparrow^3 \bar{x}_4 \uparrow^4 \bar{x}_6 \uparrow^4 \bar{x}_8 \uparrow^3 \\ & \bar{x}_{10} \uparrow^4 \omega \uparrow^1 \downarrow_3 A_6^* \bar{x}_3 \uparrow^1 \bar{x}_4 \uparrow^5 \bar{x}_6 \uparrow^6 \bar{x}_9 \uparrow^1 \bar{x}_{10} \uparrow^7 \omega \uparrow^3 \downarrow_4 A_1^* \\ & A_8^* \omega \uparrow^3 \downarrow_5 A_3 x_5 \uparrow^5 \omega \uparrow^4 \downarrow_6 A_4 x_7 \uparrow^6 \omega \uparrow^4 \downarrow_7 A_{k1}. \end{aligned} \quad (4.18)$$

В цій ЛСА 21 оператор, з яких x_4 , x_6 та x_{10} повторюються двічі. Це говорить про те, що не завжди вдається усунути повторення операторів в ЛСА.

В деяких випадках спростити ЛСА можна з урахуванням так званого розподілу зсувів. В процесі виконання алгоритму окремі функціональні оператори можуть змінювати стан логічних операторів. В цьому разі останні входять в розподіл зсуву перших [10, 18 – 20].

Нехай маємо ЛСА:

$$U = A_0 x_1 \uparrow^1 A_1 x_1 \uparrow^1 x_2 \uparrow^1 A_2 x_1 \uparrow^1 x_2 \uparrow^1 x_3 \uparrow^1 A_3 \downarrow_1 A_k.$$

з розподілом зсуву:

$$A_1 \text{ — } (x_2, x_3); \quad A_2 \text{ — } (x_3).$$

Оператор A_1 виконується тоді, коли $x_1 = 1$. Виконання A_1 та A_2 не змінює його значення, тому в подальшому перевірка значення цього оператора є зайвою і його можна вилучити з подальшої ЛСА:

$$U = A_0 x_1 \uparrow^1 A_1 x_2 \uparrow^1 A_2 x_2 \uparrow^1 x_3 \uparrow^1 A_3 \downarrow_1 A_k.$$

Оператор A_2 виконується тоді, коли $x_2 = 1$ і його виконання не змінює значення цього оператора. Тому подальшу перевірку стану x_2 можна не проводити. Остаточно ЛСА буде мати вигляд:

$$U = A_0 x_1 \uparrow^1 A_1 x_2 \uparrow^1 A_2 x_3 \uparrow^1 A_3 \downarrow^1 A_k.$$

Кількість членів ЛСА скоротилася з 9 до 6.

4.6. Формування мікрокоманд

В схемі автомата Уілкса кожному операторові ЛСА відповідає один вихід дешифратора. На практиці кожному виходу дешифратора можна поставити у відповідність не один оператор, а декілька — мікрокоманду. Склад мікрокоманди повинен бути таким, щоб при заданих логічних умовах виконувався один функціональний оператор. Це дозволяє зменшити кількість виходів дешифратора, кількість розрядів регістра команд та об'єм блоку пам'яті, тобто зменшити апаратні витрати на реалізацію автомата.

Перш ніж викласти спосіб формування мікрокоманд, введемо деякі поняття. Доданок ЛСА X_i , який виконується після X_i , називається його послідовником. Тоді X_i буде називатися попередником X_i . У будь-якого члена ЛСА може бути декілька попередників. Функціональний оператор має одного послідовника, а логічний — два послідовника, за рахунок чого в ЛСА утворюються розгалуження.

Мікрокоманда повинна складатися лише з сумісних операторів. Два функціональні оператори вважаються несумісними, якщо їх з деяких причин неможливо виконувати разом. Так, неможливо подавати на реверсивний двигун команди на рух в обох напрямках, тому що це виведе його з ладу. З точки зору техніки безпеки, заборонено подавати команду на відкриття дверей кабіни ліфта під час її руху.

Функціональний та логічний оператори несумісні, якщо останній вводить в розподіл зсуву першого, тобто, якщо виконання функціонального оператора змінює значення логічного.

Логічні оператори не можуть бути несумісними, тому що вони не виконують ніяких дій.

Група членів ЛСА утворює гілку, якщо кожний $(i + 1)$ -й член гілки є послідовником i -го члена, причому один і той же оператор може входити в декілька гілок. Група членів ЛСА утворює мікрокоманду, якщо всі її члени сумісні. Така група називається максимальною, якщо введення до неї ще одного члена робить групу несумісною.

Для складання мікрокоманд необхідно визначити сумісність операторів та розподіл зсуву, що виконується за алгоритмом роботи автомата.

Складання мікрокоманд здійснюється наступним чином:

1. Випишується перший оператор ЛСА (A_0 не враховується). Якщо це функціональний оператор, ведемо від нього стрілку і праворуч випишуємо

наступний оператор. У логічного оператора буде дві стрілки, два послідовники — верхній (при $X_i = 1$) та нижній (при $X_i = 0$).

2. Перевіряємо сумісність між послідовниками та попередниками.

3. Гілка групи закінчується, якщо до неї ввійшов останній оператор ЛСА, або, якщо наступним буде виконуватися оператор, несумісний з будь-яким оператором даної гілки.

4. Формування мікрокоманди закінчується тоді, коли обірвані всі її гілки.

5. Якщо в першу мікрокоманду ввійшли не всі оператори ЛСА, формуємо нову мікрокоманду, яка починається з оператора, який не ввійшов в попередні мікрокоманди.

6. Різні мікрокоманди можуть мати однакові гілки. Їх можна об'єднувати за допомогою стрілок з номерами. При реалізації автомата такі гілки об'єднуються за допомогою елемента “АБО”.

Нехай маємо ЛСА:

$$U = A_0 x_1 \uparrow^1 A_1 x_2 \uparrow^2 A_2 \downarrow_1 A_3 x_3 \uparrow^2 x_4 \uparrow^3 \downarrow_4 A_4 \downarrow_2 x_5 \uparrow^5 \\ \downarrow_3 A_5 A_6 x_6 \uparrow^4 A_7 \downarrow_5 A_k,$$

яка має 13 елементів.

Розподіл зсувів:

A_1 — (x_4) ; A_2 — (x_3, x_4) ; A_3 — (x_1) ; A_6 — (x_1) .

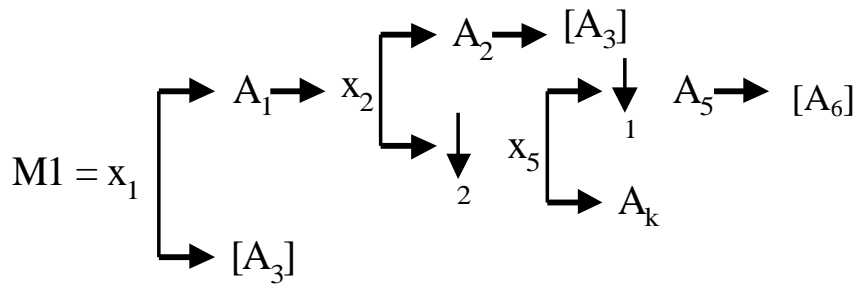
Сумісність функціональних операторів зобразимо таблицею 4.1, в якій знак “*” означає неможливість їх одночасного виконання, а знак “√” — навпаки.

Таблиця 4.1

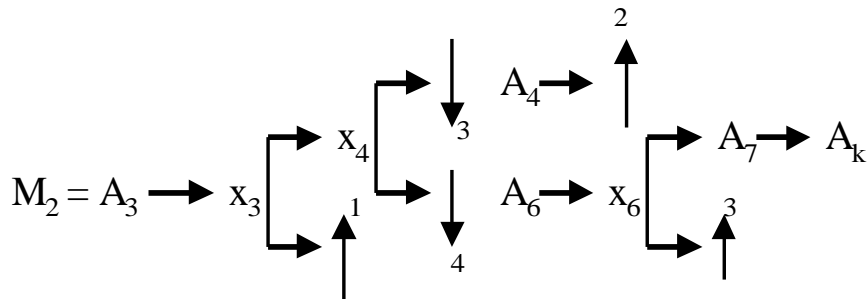
A_2	√					
A_3	*	*				
A_4	*	√	√			
A_5	√	*	√	√		
A_6	√	*	√	*	*	
A_7	*	√	√	*	√	√
	A_1	A_2	A_3	A_4	A_5	A_6

Перша мікрокоманда починається з оператора X_1 , у якого два послідовника:

Нижня гілка X_1 обривається, тому що A_3 є несумісним з ним. Оператор A_3 в мікрокоманду не входить, що показано дужками.



Друга мікрокоманда починається з оператора A_3 .



В першій мікрокоманді є оператор A_6 , який до неї не входить. Не дивлячись на те, що цей оператор входить до другої мікрокоманди, з нього необхідно починати третю мікрокоманду:

$$M_3 = \uparrow^4 .$$

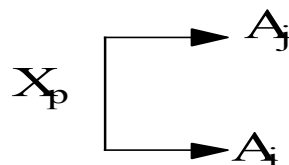
Для розподілу мікрокоманд по входах дешифратора їх необхідно закодувати. У даному прикладі прийняте таке кодування:

$$M_1 = 00, M_2 = 01, M_3 = 11.$$

Три мікрокоманди мають в своєму складі всі оператори ЛСА. Структурна схема автомата зображена на рис. 4.6.

На входах блоку пам'яті БП вказані коди мікрокоманд, які повинні бути занесені в регістр. Вільний вихід дешифратора відповідає оператору A_k .

Очікуючі оператори завжди несумісні. Враховуючи, що МПА очікує коли логічні умови приймуть відповідне значення, цю несумісність можна усунути, якщо при формуванні мікрокоманди поміняти місцями функціональний та логічний оператори. Але в цьому разі продовжувати гілку $X_p = 0$ не дозволяється.



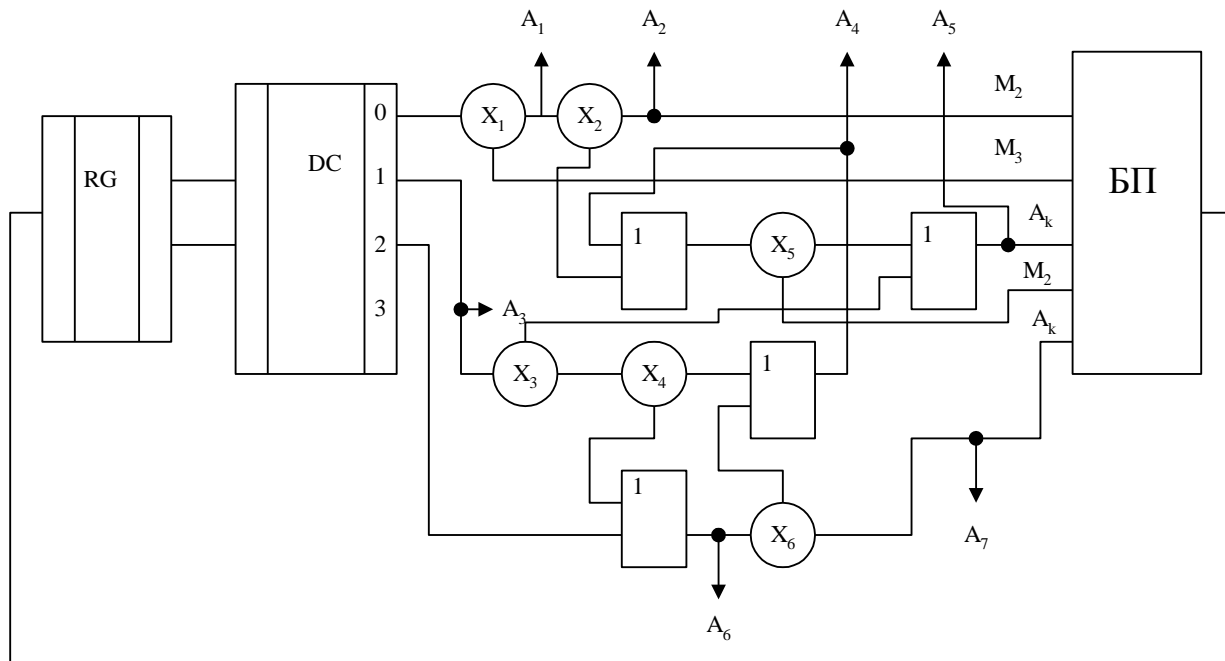


Рис. 4.6

4.7. Часткові логічні схеми алгоритмів та їх об'єднання

Описані вище МПА виконували одну програму і називалися автономними автоматами. В той же час можливо розробити автомат, який може виконувати одну з декількох програм. ЛСА, які описують кожну таку програму, називаються частковими ЛСА. Вони можуть мати деяку кількість спільних операторів, тому що мова йде про управління одним і тим же процесом, але по різних програмах.

Для того, щоб МПА міг перейти від виконання однієї програми до іншої, він повинен мати зовнішні входи управління $r_1 \div r_k$. Кількість таких входів дорівнює $\log_2 L$ (де L — кількість програм). Набір значень $\tilde{r}_1 \dots \tilde{r}_k$ є не що інше, як код даної програми.

Нехай перша програма описується ЛСА U_1 а друга — ЛСА U_2 . Тоді ЛСА, яка описує роботу автомата, має вигляд:

$$U = \downarrow_1 \bar{r} \uparrow^2 U_1 \omega \uparrow^1 \downarrow_2 U_2 \omega \uparrow^1.$$

при $r = 0$ виконується U_1 , а при $r = 1$ — виконується U_2 .

ЛСА U_1 та U_2 можуть мати спільні члени. Нехай:

$$U_1 = U_3 U_4, \quad \text{а} \quad U_2 = U_3 U_5.$$

Тоді:

$$U = \downarrow_1 U_3 \bar{r} \uparrow^2 U_4 \omega \uparrow^1 \downarrow_2 U_5 \omega \uparrow^1.$$

Якщо програми повинні переключатися автоматично, необхідно ввести оператори, один з яких встановлює r в нуль (F_0), а другий — в одиницю (F_1).

$$U = \downarrow_1 U_3 \bar{r} \uparrow^2 U_4 F_1 \omega \uparrow^1 \downarrow_2 U_5 F_0 \omega \uparrow^1.$$

Цей приклад показує, що об'єднання ЛСА приводить до появи додаткових змінних $r_1 \div r_k$, тому що МПА повинен знати, яку програму він виконує.

Існуючі алгоритми кодування часткових ЛСА [10, 18 – 20] досить складні. За цими алгоритмами дві ЛСА, які мають найбільшу кількість спільних пар $x_i A_i$, повинні кодуватися сусідніми кодами, тобто їх коди повинні відрізнятися однією змінною. В цьому разі кількість входжень додаткових змінних в загальну ЛСА буде мінімальною.

Об'єднання ЛСА здійснюється за таким правилом:

- кодуємо кожну ЛСА, тобто кожній ЛСА приписуємо певний набір змінних $r_1 \div r_k$.
- складаємо формули переходів для всіх операторів кожної ЛСА. Кожна формула переходів множиться на код своєї ЛСА.
- складаємо формули переходів загальної ЛСА. Формули переходів такої ЛСА є логічна сума однакових операторів кожної ЛСА.
- складаємо загальну ЛСА по цим формулам переходів по правилам, які були викладені раніше.

Складемо загальну ЛСА прикладу з підрозділу 4.4. Будемо вважати, що ЛСА (4.9) є U_1 і має код $r = 0$. ЛСА (4.10) є U_2 і має код $r = 1$. Будемо також вважати, що оператор A_{k1} переводить r в одиницю, а оператор A_k — в нуль.

Запишемо ці ЛСА у вигляді формул переходів:

$$A_0 \rightarrow x_1 A_1 + \bar{x}_1 A_0;$$

$$A_1 \rightarrow \bar{r} A_8;$$

$$A_8 \rightarrow \bar{r} A_6;$$

$$A_6 \rightarrow \bar{r}(x_2 A_6^* + x_4 A_1^* + x_6 A_1^* + x_8 A_6^* + x_{10} A_1^*);$$

$$A_6^* \rightarrow \bar{r}(x_3 A_6 + x_4 A_3 + x_6 A_4 + x_9 A_6 + x_{10} A_{k1});$$

$$A_1^* \rightarrow A_8^*;$$

$$A_8^* \rightarrow \bar{r} A_6^*;$$

$$A_3 \rightarrow \bar{r} x_5 A_1;$$

$$A_4 \rightarrow \bar{r} x_7 A_1;$$

$$A_{k1} \rightarrow x_{10} A_2;$$

$$A_2 \rightarrow A_8;$$

$$A_8 \rightarrow r A_6;$$

$$A_6 \rightarrow r(x_9 A_6^* + x_7 A_2^* + x_5 A_2^* + x_3 A_6^* + x_1 A_2^*);$$

$$A_6^* \rightarrow r(x_8 A_6 + x_7 A_3 + x_5 A_4 + x_4 A_6 + x_1 A_k);$$

$$A_2^* \rightarrow A_8^*;$$

$$A_8^* \rightarrow r A_6^*;$$

$$A_3 \rightarrow rx_6 A_2;$$

$$A_4 \rightarrow rx_4 A_2.$$

Через очевидність запишемо не всі формули переходів загальної ЛСА, а тільки деякі з них:

$$A_8 \rightarrow \bar{r}A_6 + rA_6 \rightarrow A_6;$$

$$A_8^* \rightarrow \bar{r}A_6^* + rA_6^* \rightarrow A_6^*;$$

$$A_3 \rightarrow \bar{r}x_5 A_1 + rx_6 A_2;$$

$$A_4 \rightarrow \bar{r}x_7 A_1 + rx_4 A_2.$$

Складемо загальну ЛСА:

$$U = \downarrow_1 A_0 x_1 \uparrow^1 \downarrow_6 A_1 \downarrow_{11} A_8 \downarrow_2 A_6 \bar{r} \uparrow^{12} \bar{x}_2 \uparrow^3 \bar{x}_4 \uparrow^4 \bar{x}_6 \uparrow^4 \\ \bar{x}_8 \uparrow^3 \bar{x}_{10} \uparrow^4 \omega \uparrow^2 \downarrow_3 A_6^* \bar{r} \uparrow^{15} \bar{x}_3 \uparrow^2 \bar{x}_4 \uparrow^5 \bar{x}_6 \uparrow^8 \bar{x}_9 \uparrow^2 \\ \bar{x}_{10} \uparrow^{10} \omega \uparrow^3 \downarrow_4 A_1^* \downarrow_{14} A_8^* \omega \uparrow^3 \downarrow_5 A_3 \bar{r} \uparrow^7 x_5 \uparrow^5 \omega \uparrow^6 \\ \downarrow_7 x_6 \uparrow^5 \omega \uparrow^{17} \downarrow_8 A_4 \bar{r} \uparrow^9 x_7 \uparrow^8 \omega \uparrow^6 \downarrow_9 x_4 \uparrow^8 \omega \uparrow^{17} \\ \downarrow_{10} A_{k1} x_{10} \uparrow^{10} \downarrow_{17} A_2 \omega \uparrow^{11} \downarrow_{12} \bar{x}_9 \uparrow^3 \bar{x}_7 \uparrow^{13} \bar{x}_5 \uparrow^{13} \bar{x}_3 \uparrow^2 \\ \bar{x}_1 \uparrow^{13} \omega \uparrow^2 \downarrow_{13} A_2^* \omega \uparrow^{14} \downarrow_{15} \bar{x}_8 \uparrow^2 \bar{x}_7 \uparrow^5 \bar{x}_5 \uparrow^8 \bar{x}_4 \uparrow^2 \bar{x}_1 \uparrow^{16} \\ \omega \uparrow^3 \downarrow_{16} A_k.$$

Загальна ЛСА має 40 операторів, замість 42 в окремих ЛСА. Таке незначне скорочення ЛСА пояснюється тим, що одні і ті ж функціональні оператори в різних ЛСА виконуються при різних логічних умовах, тобто ці ЛСА практично не мають спільних пар $x_i A_i$.

Будь-яка ЛСА описує процес управління, який виконується послідовно в часі. В реальних об'єктах досить часто механізми працюють паралельно в часі. В теорії МПА існує механізм запису паралельних ЛСА [10], але ми його не наводимо не тільки через громіздкість цього запису, а головним чином тому, що для кожної паралельної ЛСА необхідно мати свій автомат. Тому більш доцільно такі ЛСА записувати окремо одна від одної.

4.8. Практичне проектування мікропрограмних автоматів

Повну процедуру проектування МПА доцільно розглянути на прикладі.

Нехай необхідно розробити МПА для управління процесом роздроблення паперової маси. Агрегат повинен працювати за наступним алгоритмом:

- якщо тиск води, яка охолоджує мастило, в нормі;
- якщо тиск в підшипниках млина мастила в нормі;
- якщо температура мастила в нормі;
- якщо струм приводу млина в нормі;

то:

- включити головний привод млина;
- відкрити заслінку на вході млина.

Якщо хоч один з вказаних параметрів не відповідає нормі, то виключити головний привод млина, закрити заслінку на вході млина і включити сигналізацію, яка б вказувала причину зупинки. Передбачити можливість ручного пуску та зупинки автомата.

Проведемо змістовний опис операторів.

A_1 — включення приводу млина;

A_2 — включення приводу відкриття заслінки;

A_3 — включення приводу на закриття заслінки;

A_4 — включення сигналізації "Тиск води нижче норми";

A_5 — включення сигналізації "Тиск мастила нижче норми";

A_6 — включення сигналізації "Температура мастила вища за норму";

A_7 — включення сигналізації "Струм приводу вище норми";

A_i^* — виключення відповідної команди.

$x_1 = 1$ — тиск води в нормі;

$x_1 = 0$ — тиск води нижче норми;

$x_2 = 1$ — тиск мастила в нормі;

$x_2 = 0$ — тиск мастила нижче норми;

$x_3 = 1$ — температура мастила вища за норму;

$x_3 = 0$ — температура мастила в нормі;

$x_4 = 1$ — струм приводу вищий за норму;

$x_4 = 0$ — струм приводу в нормі;

$x_5 = 1$ — заслінка відкрита;

$x_5 = 0$ — заслінка не відкрита;

$x_6 = 1$ — заслінка закрита;

$x_6 = 0$ — заслінка не закрита.

Складемо ЛСА процесу управління млином:

$$\begin{aligned}
 U = & \downarrow_4 A_0 \downarrow_1 x_1 \uparrow^3 x_2 \uparrow^6 \bar{x}_3 \uparrow^8 \bar{x}_4 \uparrow^{10} A_1 \downarrow_2 A_2 \\
 & x_5 \uparrow^2 A_2^* \omega \uparrow^1 \downarrow_3 A_1^* \downarrow_5 A_3 x_6 \uparrow^5 A_3^* A_4 \omega \uparrow^4 \downarrow_6 A_1^* \\
 & \downarrow_7 A_3 x_6 \uparrow_7 A_3^* A_5 \omega \uparrow^4 \downarrow_8 A_1^* \downarrow_9 A_3 x_6 \uparrow^9 A_3^* A_6 \omega \uparrow^4 \\
 & \downarrow_{10} A_1^* \downarrow_{11} A_3 x_6 \uparrow^{11} A_3^* A_7 \omega \uparrow^4 .
 \end{aligned}$$

Ця ЛСА складалася при умові, що порушення технологічних параметрів усуває технологічний персонал, який потім включає автомат в роботу.

Проаналізуємо отриману ЛСА. В ній відсутні команди на виключення сигналізації. Її можна виключати при пуску автомата і тоді після A_0 необхідно ввести оператори $A_4^*, A_5^*, A_6^*, A_7^*$. Сигналізацію можна виключати і кнопкою "ПУСК", якою технологічний працівник включає автомат в роботу. Прийнемо останнє.

Якщо всі параметри в нормі, автомат включає оператор A_1 , потім A_2 і, після відкриття заслінки, оператор A_2^* , а потім знову переходить до перевірки

стану параметрів. При цьому знову включається A_1 , A_2 та A_2^* . Подача команди включення на пристрій, який вже включено, не має значення. Але подача команди A_2 приведе до включення контактора (реле) цього двигуна. Оскільки заслінка відкрита, слідом буде подана команда на виключення. В результаті контактор буде весь час включатися та виключатися. Кожний контактор розрахований на певну частоту включення та виключення, тому є велика вірогідність виходу його з ладу. Необхідно забезпечити подачу команд A_1 , A_2 та A_2^* тільки тоді, коли привод ще не включено. Для цього введемо ще один датчик:

$x_7 = 1$ — привод включено;

$x_7 = 0$ — привод виключено.

Згідно з ЛСА, при порушенні параметра спочатку виключається привод млина, а потім включається відповідна сигналізація. Це означає, що виключивши привод, автомат повинен визначати, по якій причині це відбулося, щоб включити відповідну сигналізацію. Більш доцільно спочатку включити сигналізацію, а потім виключати привод, що дає змогу не перевіряти ще раз стан датчиків. Нова ЛСА має вигляд:

$$U = \downarrow_5 A_0 \downarrow_1 x_1 \uparrow^3 x_2 \uparrow^6 \bar{x}_3 \uparrow^8 \bar{x}_4 \uparrow^{10} \bar{x}_7 \uparrow^1 A_1 \downarrow_2 A_2 \\ x_5 \uparrow^2 A_2^* \omega \uparrow^1 \downarrow_3 A_4 A_1^* \downarrow_4 A_3 x_6 \uparrow^4 A_3^* \omega \uparrow^5 \downarrow_6 A_5 A_1^* \downarrow_7 A_3 \\ x_6 \uparrow^7 A_3^* \omega \uparrow^5 \downarrow_8 A_6 A_1^* \downarrow_9 A_3 x_6 \uparrow^9 A_3^* \omega \uparrow^5 \downarrow_{10} A_7 A_1^* \downarrow_{11} A_3 \\ x_6 \uparrow^{11} A_3^* \omega \uparrow^5 .$$

Перейдемо до спрощення ЛСА. Об'єднаємо оператори:

$$A_1 A_2 = A_8;$$

$$A_1^* A_3 = A_9.$$

В ЛСА є очікуючі оператори:

$$A_2 x_5 A_2^* \text{ та } A_3 x_6 A_3^* .$$

Виключимо з ЛСА оператори A_2^* та A_3^* , але команди на включення не будуть мати тригерів для запам'ятовування.

Маємо нову ЛСА:

$$U = \downarrow_5 A_0 \downarrow_1 x_1 \uparrow^3 x_2 \uparrow^6 \bar{x}_3 \uparrow^8 \bar{x}_4 \uparrow^{10} \bar{x}_7 \uparrow^1 \downarrow_2 A_8 x_5 \uparrow^2 \omega \uparrow^1 \\ \downarrow_3 A_4 \downarrow_4 A_9 x_6 \uparrow^4 \omega \uparrow^5 \downarrow_6 A_5 \downarrow_7 A_9 x_6 \uparrow^7 \omega \uparrow^5 \downarrow_8 A_6 \downarrow_9 A_9 \\ x_6 \uparrow^9 \omega \uparrow^5 \downarrow_{10} A_7 \downarrow_{11} A_9 x_6 \uparrow^{11} \omega \uparrow^5 .$$

Для подальшого скорочення ЛСА, складемо формули переходів:

$$A_0 \rightarrow \downarrow_1 x_1 x_2 \bar{x}_3 \bar{x}_4 \bar{x}_7 A_8 + \bar{x}_1 A_4 + \bar{x}_2 A_5 + x_3 A_6 + x_4 A_7 + x_7 \uparrow^1;$$

$$A_8 \rightarrow x_5 \uparrow^1 + \bar{x}_5 A_8;$$

$$A_4 \rightarrow \downarrow_2 A_9;$$

$$A_5 \rightarrow A_9 = \omega \uparrow^2;$$

$$A_6 \rightarrow A_9 = \omega \uparrow^2;$$

$$A_7 \rightarrow A_9 = \omega \uparrow^2;$$

$$A_9 \rightarrow x_6 A_0 + \bar{x}_6 A_9.$$

Перейдемо від формул переходів до ЛСА:

$$U = \downarrow_4 A_0 \downarrow_1 x_1 \uparrow^4 x_2 \uparrow^5 \bar{x}_3 \uparrow^6 \bar{x}_4 \uparrow^7 \bar{x}_7 \uparrow^1 \downarrow_3 A_8 x_5 \uparrow^3 \omega \uparrow^1$$

$$\downarrow_4 A_4 \downarrow_2 A_9 x_6 \uparrow^2 \omega \uparrow^4 \downarrow_5 A_5 \omega \uparrow^2 \downarrow_6 A_6 \omega \uparrow^2 \downarrow_7 A_7 \omega \uparrow^2.$$

Скорочена ЛСА має 13 операторів. Перейдемо до формування мікрокоманд. Для цього визначимо сумісність операторів.

Сумісність функціональних операторів представимо таблицею 4.2.

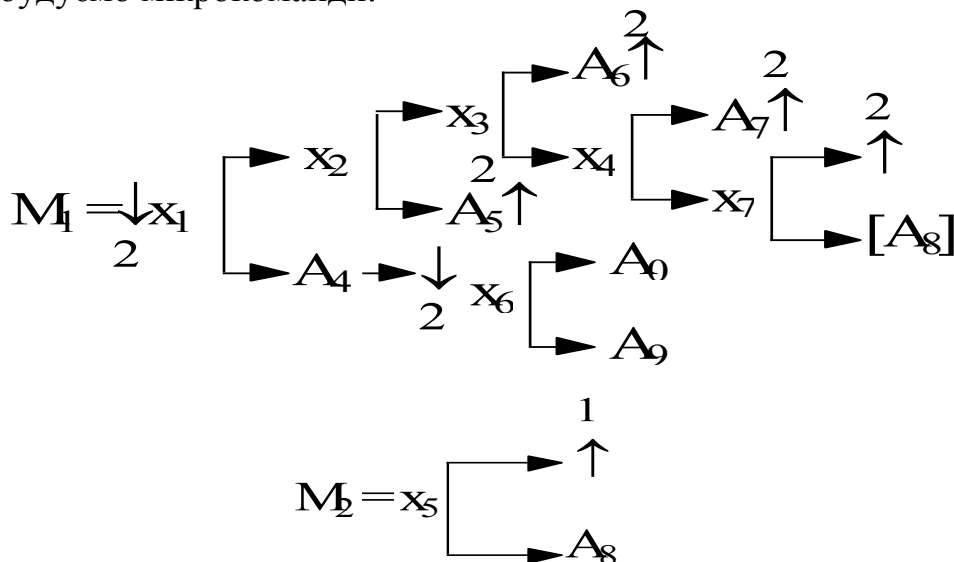
Таблиця 4.2

A_9	*				
A_4	✓	✓			
A_5	✓	✓	✓		
A_6	✓	✓	✓	✓	
A_7	✓	✓	✓	✓	✓
	A_8	A_9	A_4	A_5	i_6

Оператор A_1 змінює стан датчика x_7 . Оператори A_2 та A_3 змінюють стан датчиків x_5 та x_6 .

$$A_8 \text{ — } (x_5, x_6, x_7); \quad A_9 \text{ — } (x_5, x_6, x_7).$$

Побудуємо мікрокоманди:



Маємо дві мікрокоманди. Для того, щоб мати можливість виключати автомат, необхідно мати ще один вихід дешифратора, який не з'єднаний зі схемою автомата. Коди команд:

$$M_1 = 01; \quad M_2 = 10; \quad M_{\text{вук}} = 00.$$

Структурна схема автомата зображена на рис. 4.7.

Для переходу до функціональної та принципової схеми необхідно вибрати елементи, на яких буде реалізовано автомат.

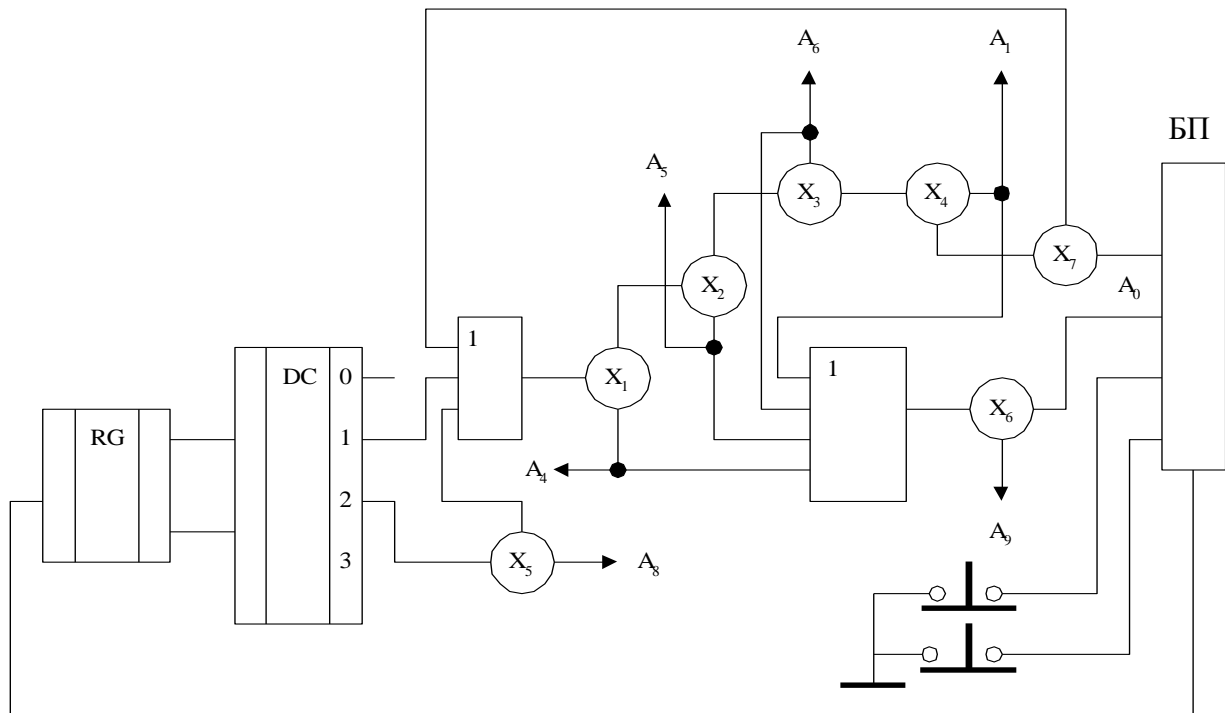


Рис. 4.7

При виборі типу регістра необхідно мати на увазі, що при виконанні чергової команди сигнали на вході БП дорівнюють нулю і тільки при видачі наступної команди на одному з входів БП з'являється одиничний сигнал. В цей момент часу на виході БП з'явиться код наступної команди, після чого сигнали на виході блоку пам'яті знову приймуть нульове значення. Це говорить про те, що регістри в інтегральному виконанні для МПА не підходять, тому що в них є синхровхід, на який поступають сигнали від генератора імпульсів. Зміна команд в МПА визначається станом об'єкта управління і синхронізувати його генератором імпульсів неможливо. Можна об'єднати входи БП елементом АБО, вихід якого може грати роль синхронізатора регістрів. Більш доцільно реалізувати регістр на окремих *RS*- або *JK*- тригерах, зміна стану яких будуть здійснювати вихідні сигнали блоку пам'яті. Дешифратор *DC* можна вибрати в інтегральному виконанні. Якщо він має інверсні виходи, то це треба враховувати при реалізації автомата.

Нехай для реалізації автомата з наведеного прикладу вибрані елементи ТТЛ серії. В цій серії немає *RS*-тригера, тому реалізуємо регістр на *JK*-тригерах.

Дешифратор в ТТЛ-серії має інверсні виходи. Тоді, згідно правилу де Моргана, елемент "АБО" (на рис. 4.7) необхідно замінити на елемент "І-НІ". Вузол перевірки стану датчика x_5 , який в загальному вигляді складається з двох елементів "І", необхідно замінити на два елементи "АБО-НІ", причому

вихід “ТАК” буде на тому елементі, на вхід якого подається \bar{x}_5 . Решта елементів схеми залишається незмінною.

Вихідний блок буде мати тригери для запам’ятовування вихідних команд МПА (рис. 4.8).

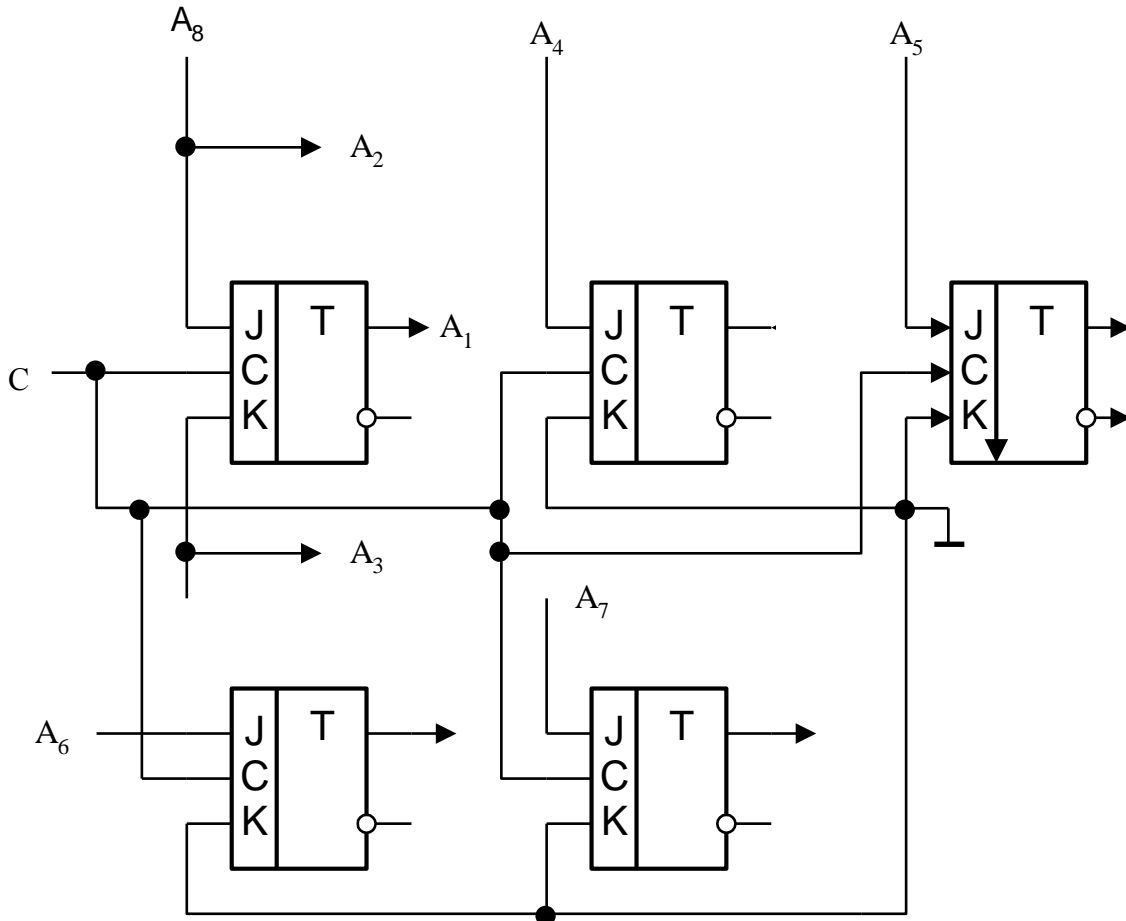


Рис. 4.8

Блок пам’яті можна реалізувати на постійних запам’ятовуючих пристроях, але їх об’єм буде використовуватися нераціонально, тому що їх вхідний сигнал є код “один з ...”. По суті БП є перетворювач коду “один з...” в код мікрокоманд, тобто є звичайною багатовихідною схемою.

Для наочності представимо блок пам’яті у вигляді діодної матриці “АБО” (рис. 4.9), в якій темними кружками показані діоди, які з’єднують відповідні горизонтальні та вертикальні шини.

По діодній матриці можна записати рівняння виходів блоку пам’яті:

$$J_1 = M_2;$$

$$K_1 = A_0 + \Pi + C;$$

$$J_2 = \Pi;$$

$$K_2 = M_2 + A_0 + C.$$

Відповідна цим рівнянням схема зображена на рис. 4.10.

Апаратна реалізація МПА вимагає великої кількості елементів, особливо при великій кількості логічних умов (датчиків). Більш доцільно

реалізувати автомат на базі мікропроцесорної системи (МПС). В цьому разі немає потреби формувати мікрокоманди. Програму роботи МПС складають на основі скороченої ЛСА.

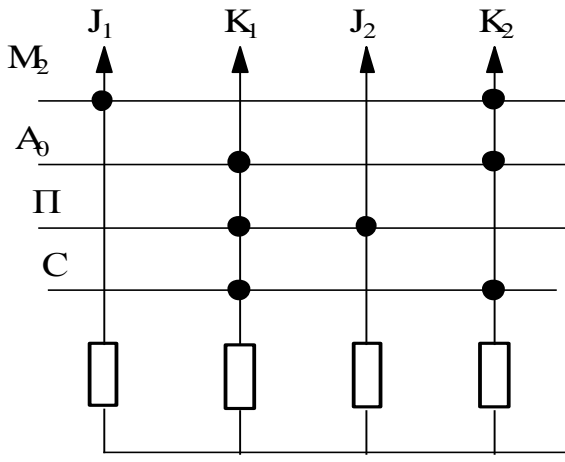


Рис. 4.9

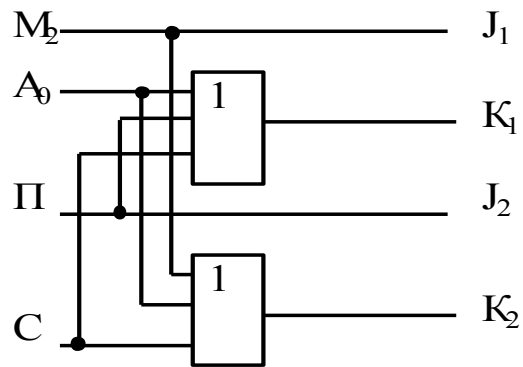


Рис. 4.10

Для підключення датчиків та вихідних сигналів МПС повинна мати вхідний порт (наприклад, порт А) та вихідний порт (наприклад, порт Б).

Для задачі, яка розглядалася раніше, структурна схема МПС буде мати вигляд (рис. 4.11).

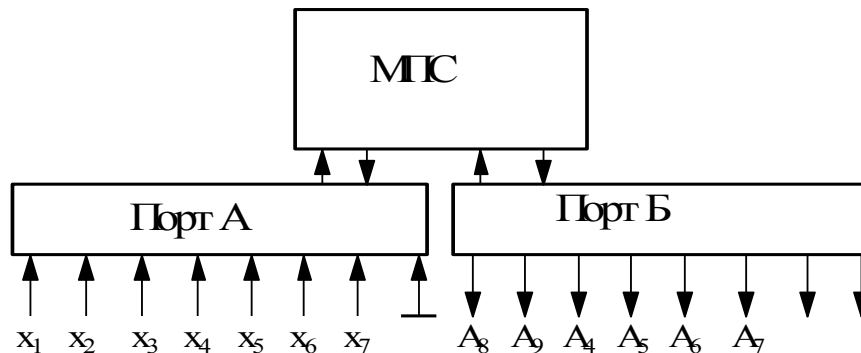


Рис. 4.11

Програма роботи МПС по ЛСА (4.19) на мові Асемблер для К580ИК80А має вигляд:

```

n1 IN A
  ANA 01H перевірка стану кнопки "ПУСК"
  JZ n2
  JMP n1
n2 IN A
  ANA 80H перевірка стану датчика x1
  JZ n3
  MOV A, 20H
  QUT Б   видача команди A4
n5 MOV A, 40H
  QUT Б   видача команди A9
  
```

n_4 IN A
 ANA 04H перевірка стану датчика x_6
 JNZ n_4
 JMP n_1

n_3 IN A
 ANA 40H перевірка стану датчика x_2
 JZ n_6
 MOV A, 10H
 QUT Б видача команди A_5
 JMP n_5

n_6 IN A
 ANA 40H перевірка стану датчика x_3
 JNZ n_7
 MOV A, 04H
 QUT Б видача команди A_6
 JMP n_5

n_7 IN A
 ANA 10H перевірка стану датчика x_4
 JNZ n_8

4.9. Контрольні питання до розділу 4

1. Дайте визначення мікропрограмного автомату (МПА).
2. Послідовність кроків по розробці МПА.
3. Які форми представлення алгоритмів роботи МПА існують?
4. Які блоки входять до структурної схеми МПА?
5. Дайте визначення логічної схеми алгоритму (ЛСА).
6. Методика складання ЛСА.
7. Назвіть види операторів в ЛСА.
8. Як можна спростити ЛСА?
9. Назвіть основні правила формування мікрокоманд.
10. Як реалізувати схему блоку пам'яті МПА?

РОЗДІЛ 5

ПРИКЛАДИ РОЗРОБКИ СТРУКТУРНОЇ ТА ПРИНЦИПОВОЇ СХЕМИ СИСТЕМИ УПРАВЛІННЯ

5.1. Розробка структурної схеми системи управління

Якщо систему управління необхідно реалізувати на базі цифрових елементів, то її структурну схему складають на основі системи рівнянь включення та виключення механізмів технологічної лінії.

Основу схеми складають тригери управління, кількість яких дорівнює кількості пускових пристроїв, тому що реверсивні механізми мають по два таких пристрою, для управління рухом механізму як в прямому, так і в зворотному напрямку. Якщо в рівняннях є додаткові тригери, їх також зображають на схемі, але виходи цих тригерів використовуються лише для управління основними.

Кожне рівняння включення або виключення є кон'юнкцією певної кількості змінних, які об'єднують за допомогою елемента "І". Вихід цього елемента подається на *S*- або *R*- входи відповідного тригера. Якщо механізм має декілька режимів включення (виключення), то будемо мати декілька відповідних рівнянь, які об'єднуються елементом "АБО".

На структурній схемі зображують датчики, виходи яких подаються на вхід елементів через перетворювачі сигналів. При реалізації системи ці перетворювачі можуть бути відсутніми, якщо датчики видають сигнали, з якими можуть працювати відповідні елементи. Виходи тригерів подають на пускові пристрої механізмів через підсилювачі, які також зображають на структурній схемі.

Для системи управління з розділу 2 (див. рис. 2.3), яка описана рівняннями (2.1 - 2.14), структурна схема має вигляд, наведений на рис. 5.1.

Вихідні сигнали датчиків $X_1 \div X_7$ повинні мати рівень, який визначається типом вибраних елементів. Тому, можливо, необхідно мати перетворювачі цих сигналів Π_i . Вихідні сигнали тригерів мають малу потужність і не можуть включити пускові пристрої механізмів. Тому необхідно мати підсилювачі сигналів ПС_i , коефіцієнт підсилення яких визначається відношенням струму ПП_i до вихідного струму тригерів.

Схема на рис. 5.1 є структурною схемою системи управління. Для переходу до принципової схеми необхідно перш за все вибрати конкретні тригери. Якщо ці тригери управляються нульовими сигналами (наприклад, ТТЛ – серія), то отримані рівняння слід переробити за законами математичної логіки.

Якщо система управління розробляється на базі мікропрограмного автомата, то основою для структурної схеми є система мікрокоманд.

Відомо [4], що структурна схема мікропрограмного автомата складається з реєстра команд(РК), дешифратора (ДС), вихідного блоку, блоку перевірки логічних умов (БПЛУ) та програмного блоку.

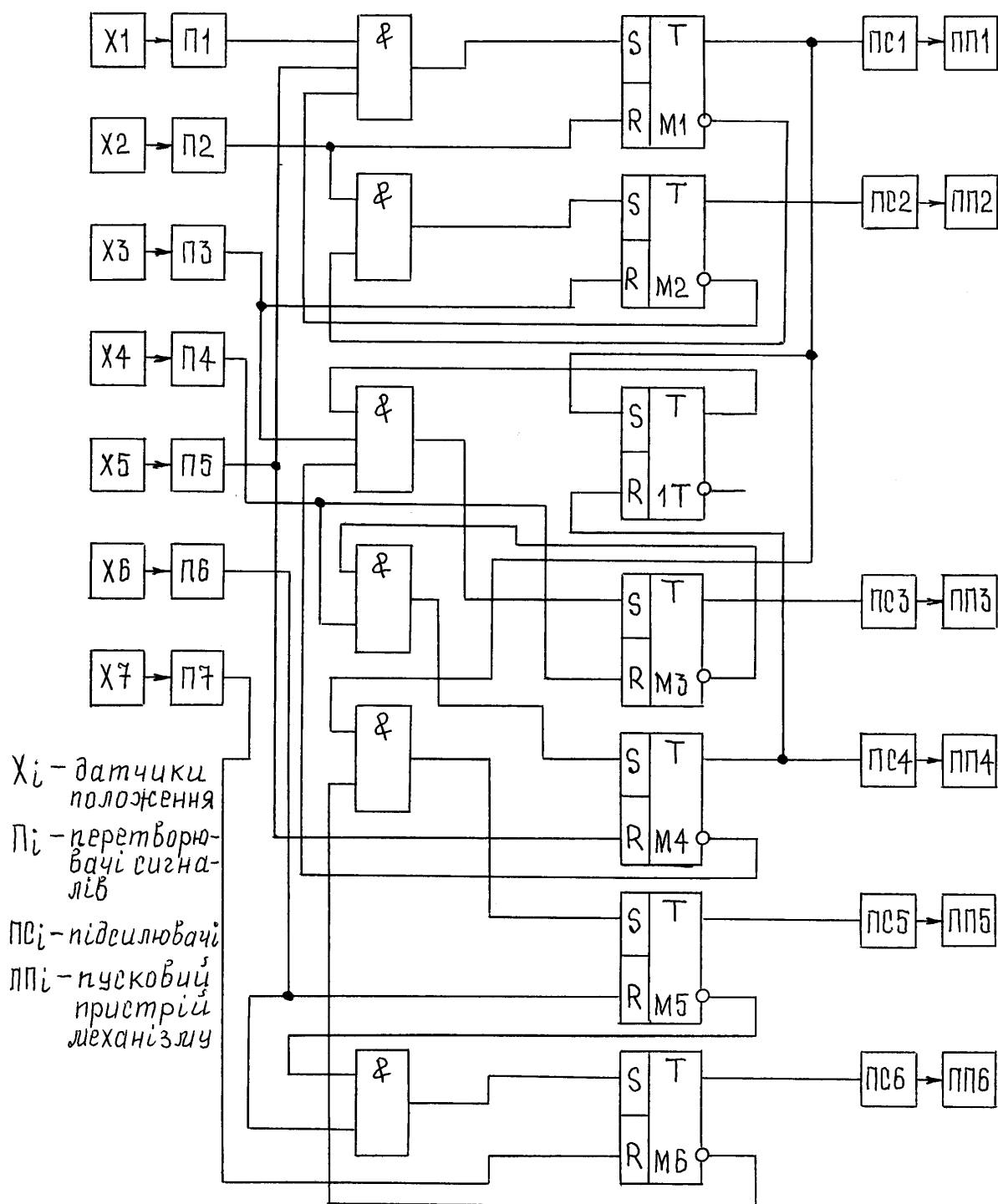


Рис. 5.1. Структурна схема системи управління

Кожній мікрокоманді відводиться один вихід дешифратора. Крім того, необхідно передбачити ще один додатковий вихід дешифратора, який не з'єднується з іншими елементами і який необхідний для виключення автомата.

Кількість виходів дешифратора визначає і кількість його входів:

$$A = \log_2(N + 1), \quad (5.1)$$

де A – кількість входів дешифратора; N – кількість мікрокоманд.

Входи дешифратора є виходами регістра команд. Рекомендується реалізувати регістр команд на асинхронних RS – тригерах, управління якими здійснює блок пам'яті.

Вихідний блок містить в собі тригери, які запам'ятовують відповідні команди і виходи яких через підсилювачі поступають на пускові пристрої механізмів. Якщо команди запам'ятовувати не потрібно, то тригер відсутній і команда поступає безпосередньо на підсилювач.

Блок перевірки логічних умов складається з елементів “І”, по два елементи на один датчик. Ці елементи зображені в мікрокомандах і переносяться на структурну схему без змін. До них треба додати ще перетворювачі сигналів, які можуть стояти на виході датчика.

Блок пам'яті розробляється як комбінаційна схема типу “один з..”. Ця схема має певну кількість входів, яка визначається структурою мікрокоманд. До входів блоку пам'яті включають також кнопки “ПУСК” та “СТОП”, які виконуються з нормально замкненими контактами. Один вивід кнопки з'єднують з блоком пам'яті, другий – заземляють. Кількість виходів блоку пам'яті дорівнює сумі R - та S - входів тригерів регістра.

Згідно з принципом дії мікропрограмного автомата, одиничний сигнал може з'явитися лише на одному з входів блоку пам'яті в той момент, коли необхідно змінити мікрокоманду. Відповідно до цього на виході блоку пам'яті з'являється код, який переключає тригери регістра і відповідно виходи дешифратора. Блок пам'яті на структурній схемі зображується у вигляді “чорного ящика” і його внутрішня структура визначається при розробці принципової схеми.

Приклад розробки структурної схеми мікропрограмного автомата наведений на рис. 5.2 згідно ЛСА (2.17), яка зводиться до двох мікрокоманд (див. розділ 2). Дешифратор мікропрограмного автомата повинен мати три виходи і два входи. Це означає, що регістр команд буде мати два виходи. Вихідний блок не буде мати тригерів, тому що команди M_i не запам'ятовуються.

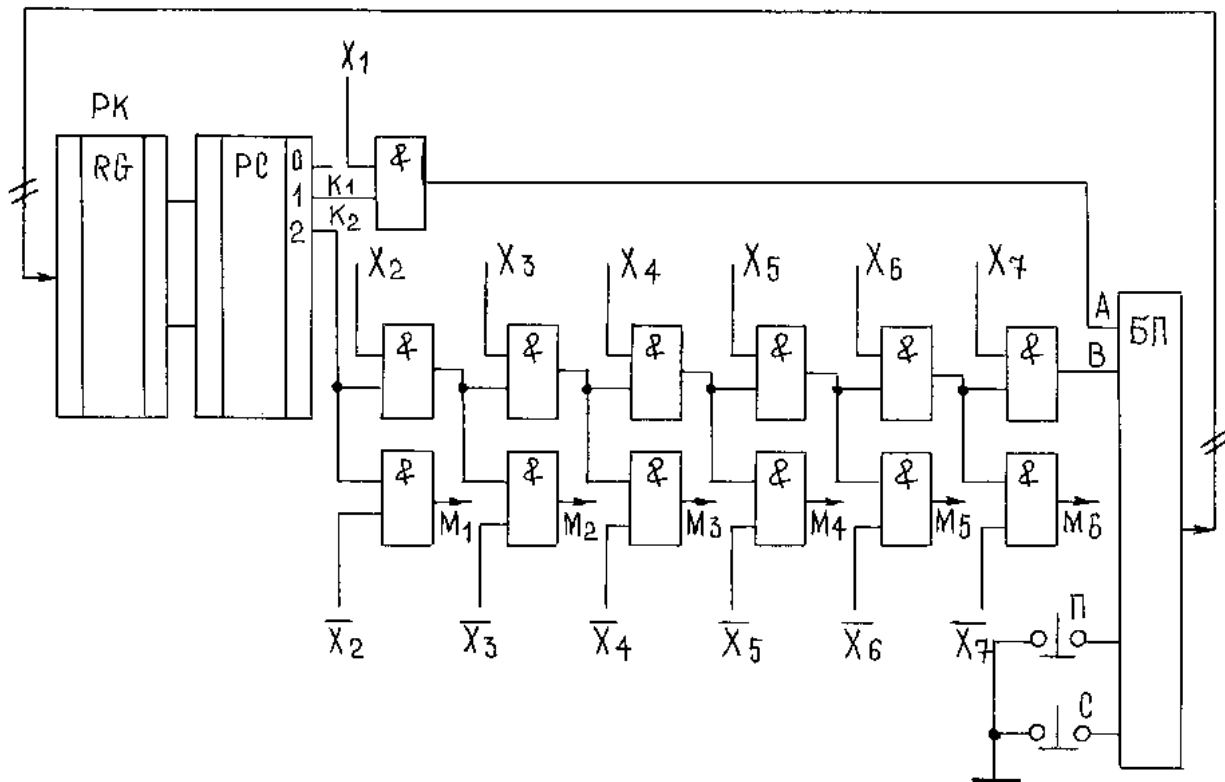
Вихідні сигнали датчиків $X_1 \div X_7$ повинні мати рівень, який визначається типом вибраних елементів. Тому, можливо, необхідно мати перетворювачі цих сигналів P_i .

Вихідні сигнали елементів мають малу потужність і не можуть включити пускові пристрої механізмів. Тому необхідно мати підсилювачі сигналів $ПС_i$, коефіцієнт підсилення яких визначається відношенням струму $ПП_i$ до вихідного струму елементів.

Схема на рисунку 5.2 є структурною схемою системи управління. Для переходу до принципової схеми необхідно перш за все вибрати конкретні елементи. Приклад розробки блоку пам'яті для даного мікропрограмного автомата буде наведена далі.

Якщо система управління реалізується на базі мікропроцесора, то її структурна схема подібна до будь-якої мікропроцесорної системи. Вона

складається з мікропроцесора та двох портів: вхідного і вихідного. На вхідний порт поступають сигнали з датчиків (можливо через перетворювачі). З вихідного порту поступають сигнали на включення механізмів і які через підсилювачі подаються на пускові пристрої механізмів. Тригери для запам'ятовування команд непотрібні, тому що вони є в вихідних портах мікропроцесорної системи.



PK – реєстр команд; ДС – дешифратор; БП – блок пам'яті;
П – кнопка “ПУСК”; С – кнопка “СТОП”.

Рис. 5.2. Структурна схема мікропрограмного автомата

Структурна схема системи управління на мікропроцесорі для технологічної ділянки по рис. 2.3 наведена на рис. 5.3. Кількість вхідних портів на ній визначається кількістю датчиків, а кількість вихідних – кількістю механізмів. На цій схемі не показані перетворювачі вихідних сигналів датчиків і підсилювачі вихідних сигналів мікропроцесора.

Вирішення рівнянь управління для кожного механізму доцільно розпочати з визначення стану цього механізму. Якщо він виключений, то немає необхідності вирішувати рівняння його виключення, тому що вони не можуть змінити стан механізму. Якщо механізм включений, то вирішуються лише рівняння виключення. Такий прийом не зменшує довжини програми, але значно скорочує час на її виконання.

Роль тригерів в цьому випадку виконують комірки оперативної пам'яті. Якщо дорівнює одиниці хоч одне рівняння включення механізму, в цю комірку записують одиницю і передають в вихідний порт мікропроцесора.

Якщо дорівнює одиниці хоча б одне рівняння виключення механізму, в цю комірку записують нуль, що виключає механізм. Якщо всі рівняння управління даним механізмом дорівнюють нулю, то стан комірки пам'яті не змінюють, і Для мікропроцесорних систем необхідно розробити блок-схему алгоритмів управління. Її структура залежить від того, як було проведено формалізацію алгоритму управління.

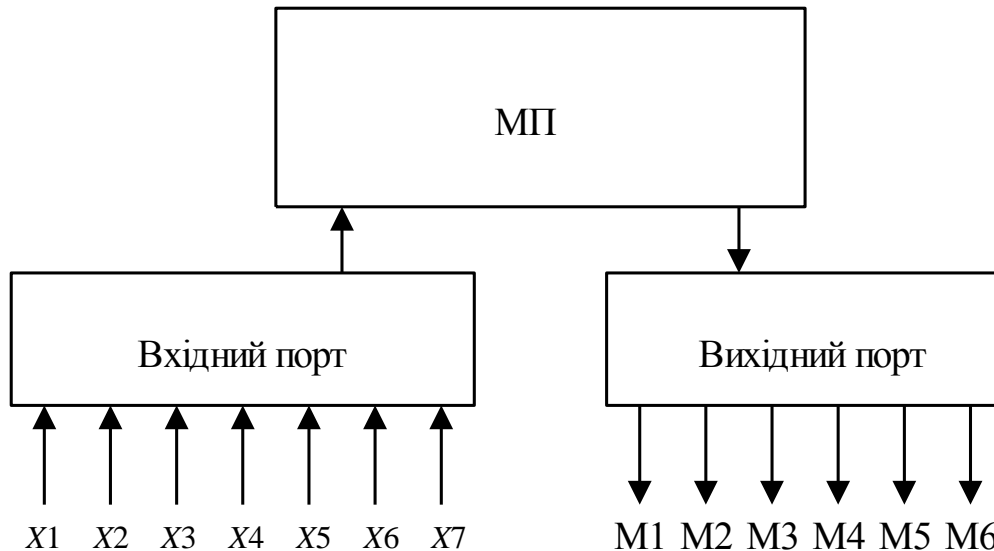


Рис. 5.3. Структурна схема мікропроцесорної системи

Якщо алгоритм управління отримано на основі системи рівнянь включення та виключення механізмів, то блок-схема алгоритму являє собою безперервний цикл вирішення цих рівнянь. Цей цикл після ініціалізації мікропроцесора починають з визначення стану кнопки “ПУСК”. Вирішення системи рівнянь починають тоді, коли ця кнопка замкнена. В кінці циклу необхідно перевірити стан кнопки “СТОП”. Якщо вона замкнена, мікропроцесор переходить в режим перевірки стану кнопки “ПУСК”. Якщо ні – мікропроцесор продовжує вирішення рівнянь управління. При такому алгоритмі зупинку системи управління можна виконати тоді, коли закінчено повний цикл управління.

мікропроцесор переходить до вирішення рівнянь управління наступного механізму. Рівняння включення та виключення механізму одночасно дорівнювати одиниці не можуть, інакше вони складені невірно.

В такій блок-схемі мікропроцесор не чекає, коли механізм виконає свою операцію, а весь час слідкує за станом датчиків і видає необхідні команди. Враховуючи велику швидкість мікропроцесора, можна вважати, що контроль за роботою механізмів буде надійним.

Приклад розробки блок-схеми алгоритмів по системі рівнянь управління (2.1 ÷ 2.14) наведений на рис. 5.4.

Якщо закон управління заданий у вигляді ЛСА, розробка блок-схеми алгоритмів по ній не є складною задачею. Кожний функціональний оператор

в ЛСА є команда на включення або виключення механізму, яку необхідно передати в вихідний порт мікропроцесора. Кожний логічний оператор в ЛСА є звернення до відповідного вхідного порту для визначення стану того чи іншого датчика. В залежності від цього мікропроцесор приймає рішення про наступну команду відповідно до ЛСА. На відміну від попереднього, при роботі по ЛСА мікропроцесор чекає, коли механізм виконає ту чи іншу операцію і лише після цього переходить до управління іншим механізмом.

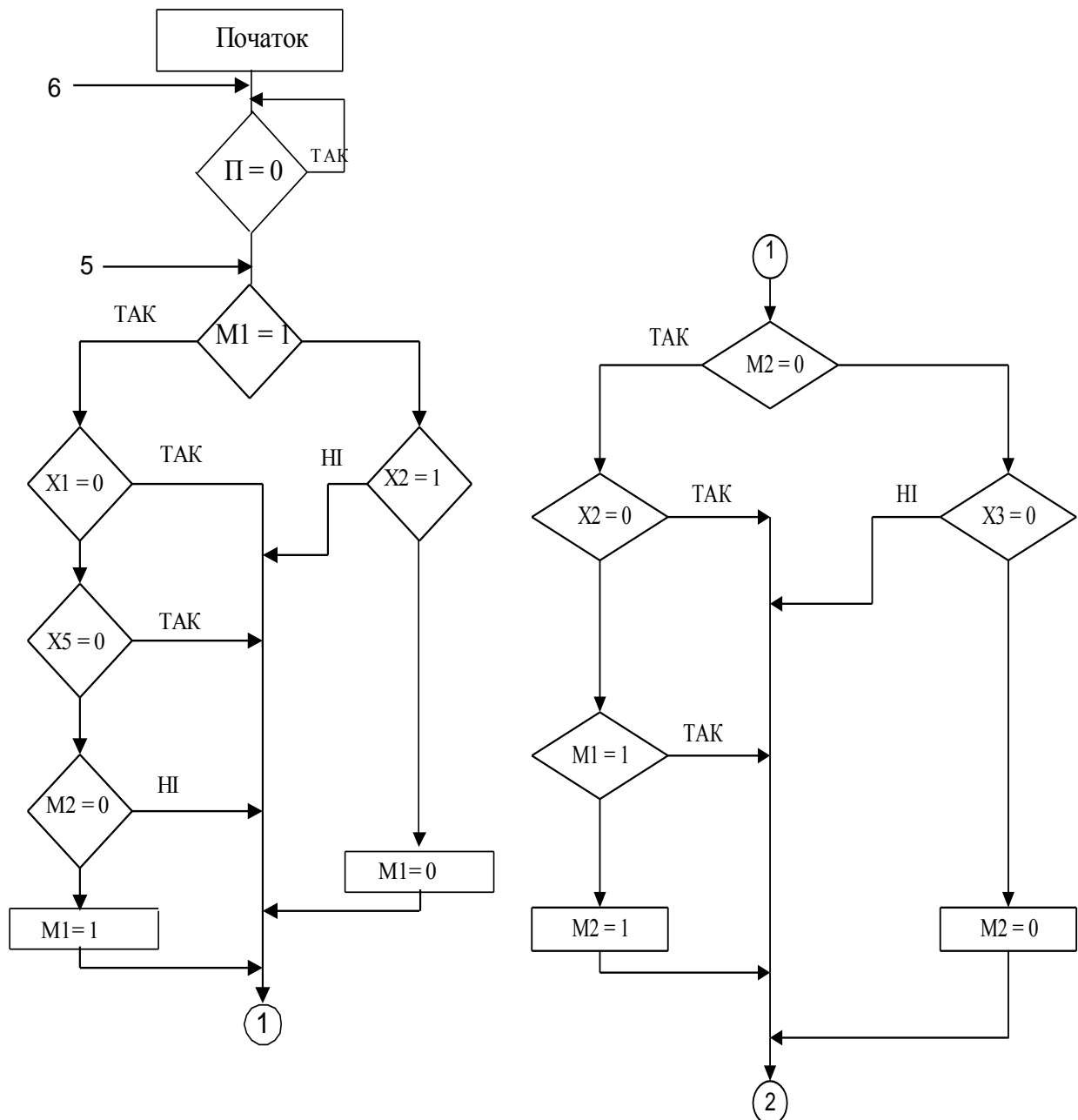


Рис. 5.4. Блок – схема алгоритму роботи системи управління

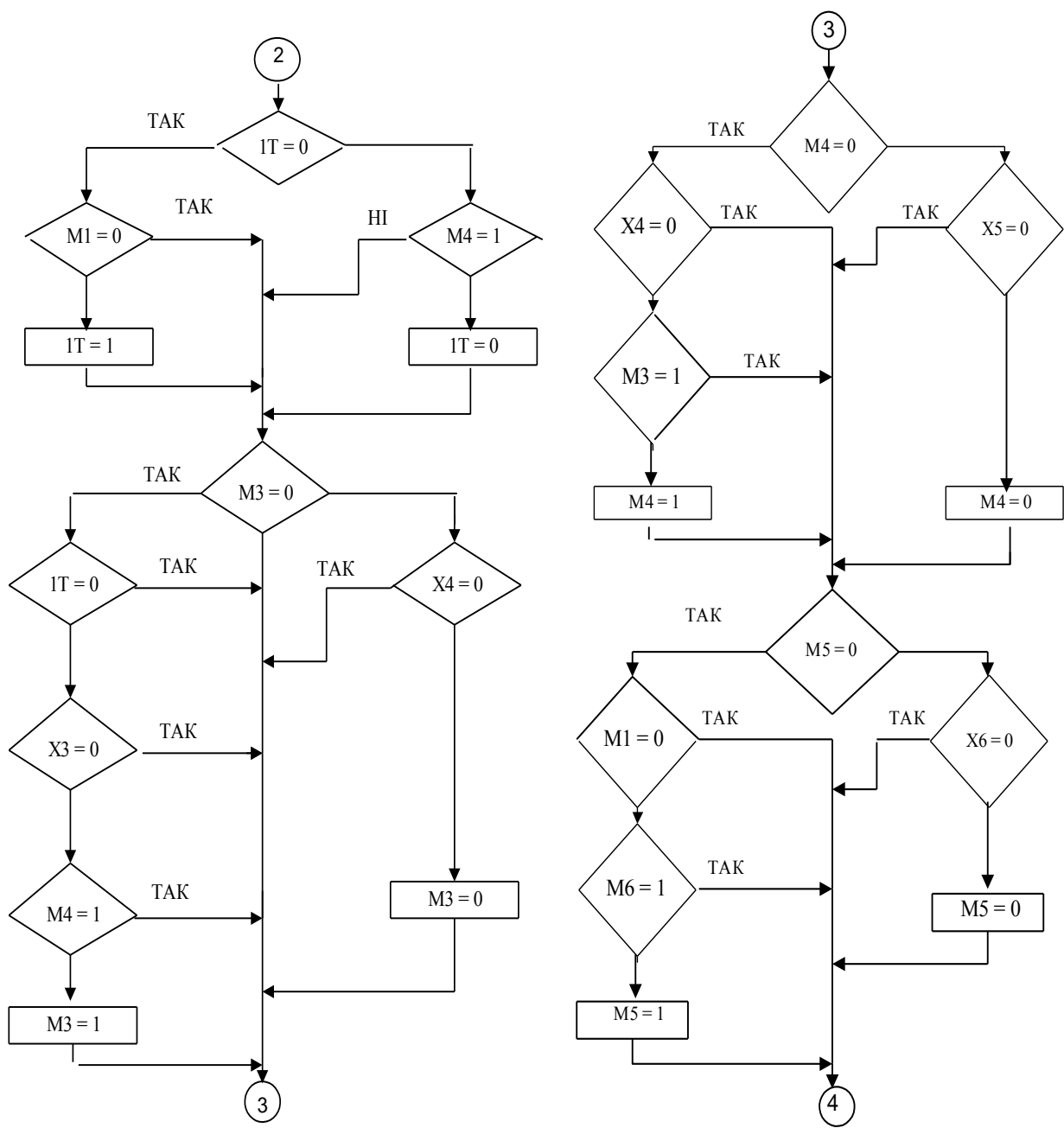


Рис. 5.4. Блок – схема алгоритму роботи системи правління (продовження)

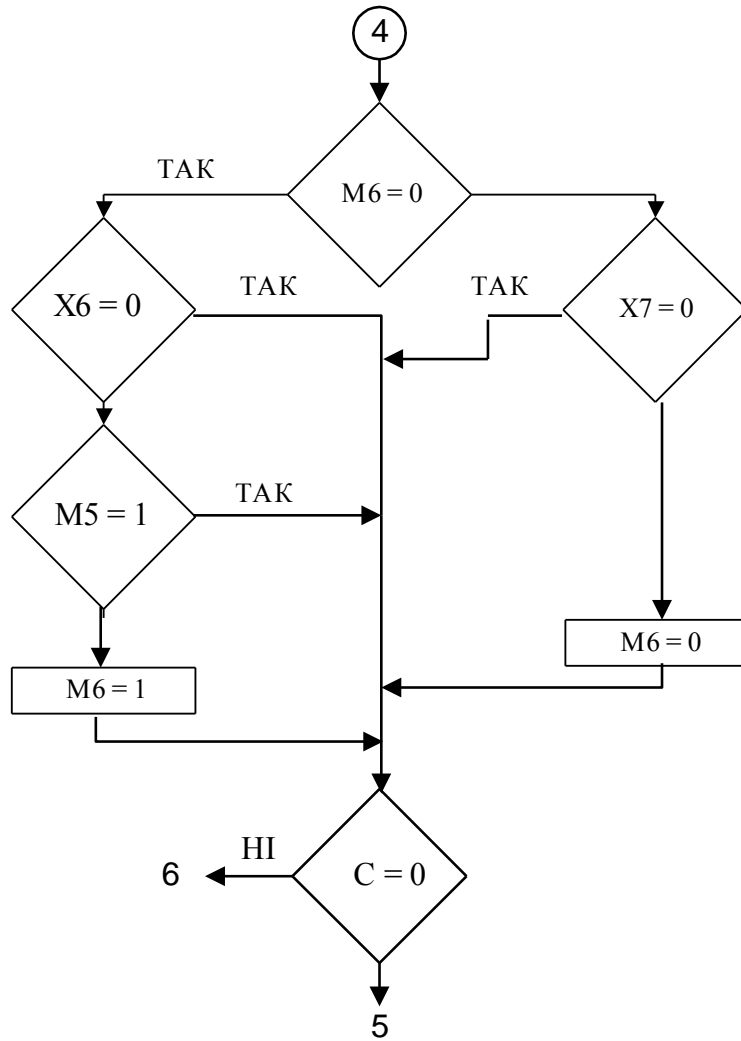


Рис. 5.4. Блок – схема алгоритму роботи системи управління (продовження)

В ЛСА необхідно ввести кнопки “ПУСК” та “СТОП”, які в мікропрограмному автоматі вводяться в блок пам’яті. Нова ЛСА, отримана з ЛСА (2.17), має вигляд:

$$\begin{aligned}
 U = & \downarrow_1 \Pi \uparrow^1 \downarrow_2 X_1 \uparrow^2 \downarrow_3 M1 \cdot X_2 \uparrow^3 M1^* \\
 & \downarrow_4 M2 \cdot X_3 \uparrow^4 M2^* \downarrow_5 M3 \cdot X_4 \uparrow^5 M3^* \\
 & \downarrow_6 M4 \cdot X_5 \uparrow^6 M4^* \downarrow_7 M5 \cdot X_6 \uparrow^7 M5^* \\
 & \downarrow_8 M6 \cdot X_7 \uparrow^8 M6^* \cdot C \uparrow^2 \omega \uparrow^1.
 \end{aligned} \tag{5.2}$$

Блок – схема алгоритмів, яка розроблена по ЛСА (5.2), наведена на рис. 5.5.

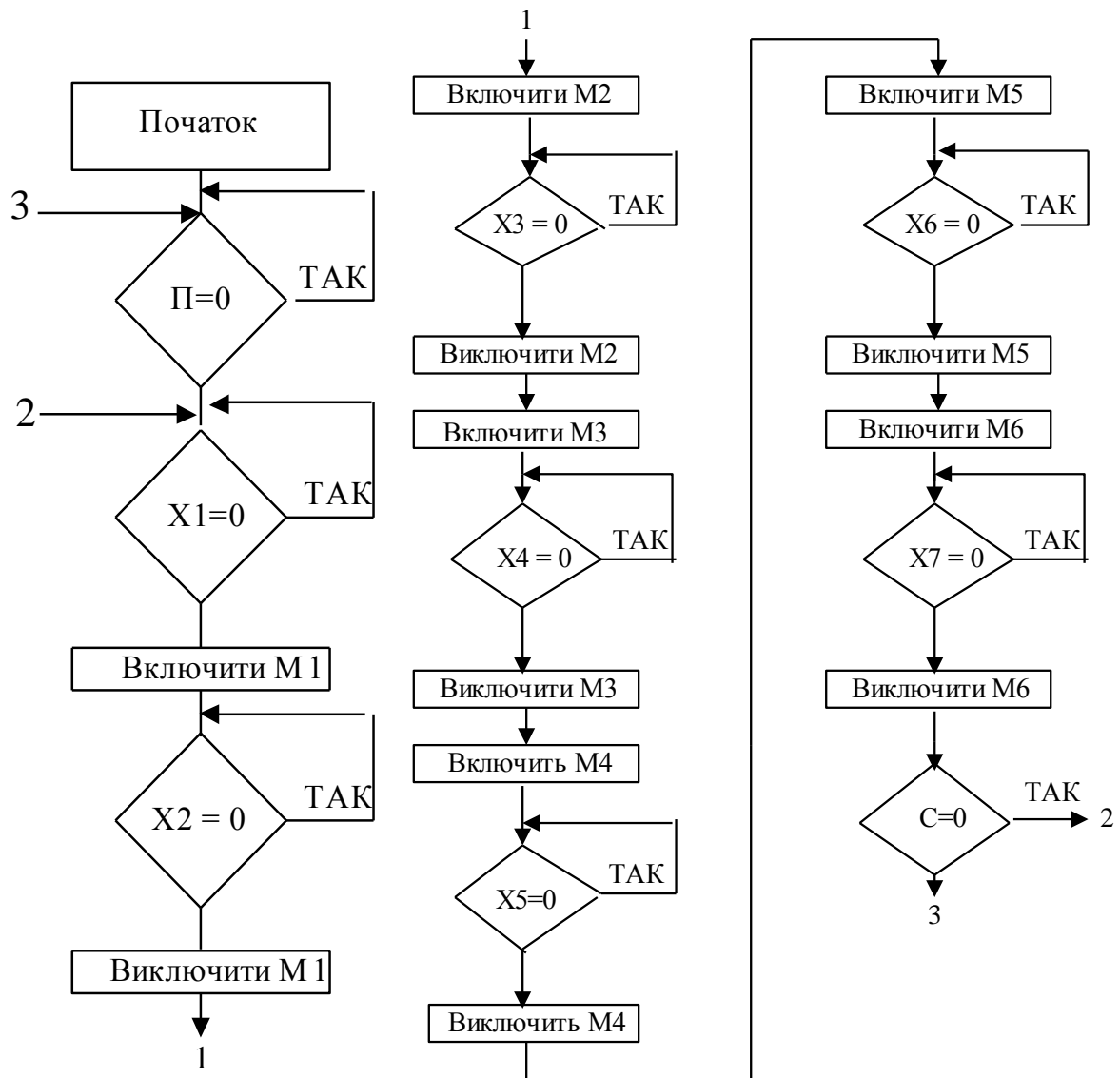


Рис. 5.5. Блок – схема алгоритму роботи системи управління згідно ЛСА (5.2)

5.2. Розробка принципової схеми системи управління

Розробку принципової схеми починають з вибору елементної бази. Принципово можливо застосування мікросхем серій ТТЛ (ТТЛШ), наприклад, К555, К1533 або мікросхем МОП (КМОП), наприклад, К561, К1564 тощо. Кожна з них має свої недоліки та переваги. Елементи серії ТТЛ мають відносно велику потужність споживання, але й велику швидкодію. Елементи серії МОП, навпаки, мають малу потужність споживання і малу швидкодію.

Швидкість переміщення робочих органів механізмів невелика. Найбільш швидкодіючі механізми мають час переміщення не менше ніж (0,4-0,5) с, а часто і більший. Це великий час в порівнянні з часом переключення елементів. З цієї точки зору вибір серії мікросхем не має значення.

В системі управління найбільшу потужність споживають елементи управління пусковими пристроями механізмів. На їх фоні потужність

споживання логічних (цифрових) елементів невелика. Тому вибір серії мікросхем визначається перш за все функціональною повнотою, тобто набором елементів, які входять до цієї серії.

При виборі ТТЛ серії слід пам'ятати, що їх *RS*-тригери управляються не одиничними, а нульовими сигналами. Дешифратори цих серій мають не прямі, а інверсні виходи. Але правила математичної логіки дозволяють зробити необхідні перетворення отриманих раніше рівнянь управління і розробити схему, в якій буде практично така ж кількість елементів.

Вибір елементів проводиться по відповідних довідниках, наприклад, [5].

В системах, які реалізуються на базі мікропроцесорної системи, головним є вибір типу мікропроцесора. Мікропроцесори типу КР580 та КР1810 сьогодні застосовувати недоцільно, тому що вони вимагають зовнішніх елементів входу/виходу, зовнішніх елементів оперативної та постійної пам'яті та зовнішніх тактових генераторів. В наш час випускається достатня кількість мікропроцесорів, які всі ці елементи мають в своєму складі (на одному кристалі), наприклад, мікропроцесори типу МК51, МК52 [6]. З них треба вибрати такий, який забезпечує необхідні операції управління і має мінімальну вартість.

Після вибору елементів необхідно вибрати датчики. В більшості випадків це датчики положення робочих органів механізмів або положення технологічного матеріалу. При цьому треба мати на увазі, що ці датчики розташовані на обладнанні або на механізмах, тобто працюють в досить важких умовах, серед пилу, бруду та вібрацій. В той же час вони повинні мати велику надійність, тому що їх пошкодження може призвести до пошкодження механізмів. Тому треба вибрати безконтактні датчики. Найбільш надійними серед них є індуктивні датчики положення типу КВД, ПЩ тощо. Вони складаються з генератора і підсилювача. При введенні в щілину між котушками генератора металевої пластини відбувається зрив генерації, що призводить до відкриття вихідного транзистора. Цей транзистор має відкритий колектор, що дозволяє отримати вихідний сигнал необхідного рівня. Якщо колектор приєднати (через резистор) до джерела напруги 5В, то можна отримати вихідний сигнал рівня ТТЛ – логіки.

Крім датчиків положення, в системі управління можуть знадобитися датчики інших параметрів: температури, тиску, витрат газу чи повітря (води), ваги тощо. Їх вибирають з довідників, виходячи з таких міркувань:

- верхня межа параметра, який вимірюється;
- чутливість;
- параметри середовища, в якому працює датчик;
- вартість датчика.

Верхня межа показань датчика повинна бути на (20-30)% більшою за максимальне значення параметра в об'єкті. Це гарантує датчик від пошкодження, якщо параметр в об'єкті досягне свого максимального значення.

Дуже важливим є середовище, в якому знаходиться датчик і яке може призвести до його пошкодження. Тому не виключено застосування спеціальної арматури для захисту датчика від впливу середовища. Це призводить до збільшення запізнення, але іншого виходу немає.

Як правило, вихідний сигнал датчиків не відповідає вимогам логічних елементів, тому застосовують перетворювачі сигналів, вихідний сигнал яких відповідає такому ряду:

0 – 5мА; 0 – 10 мА; 0 – 5В; 0 – 10В; 4 – 8 кГц.

Такі датчики мають на виході аналоговий сигнал. Якщо необхідно визначити певний рівень цього сигналу, то застосовують компаратори, в якому вихідний сигнал з'являється тоді, коли рівень вхідного дорівнює заданому. При необхідності перетворення аналогового сигналу в цифровий застосовують аналого-цифрові перетворювачі (АЦП).

В розділі “Розробка принципової схеми” необхідно вибрати спосіб управління пусковим пристроєм механізмів. Як говорилося раніше, це контактори або електро-, гідро-, пневмоклапани, тобто індуктивне навантаження, яке працює під напругою 220 або 380 В змінного струму і споживає струм від 0,4 до 1А. Застосування для таких пристроїв електромагнітних реле недоцільно через їх малу надійність. Більш доцільно застосування силових напівпровідникових елементів, серед яких найбільшою популярністю користуються симістори (симетричні тиристори) типу ТС. Ще краще застосувати оптронні симістори типу ТСО, в яких існує гальванічна розв'язка між управляючою та силовою частинами. В мережі 220В амплітудне значення напруги досягає 384В, тому симістори необхідно вибрати на напругу в 400В. Симістор типу ТСО-5-400 може витримати струм до 5А, що достатньо для більшості пускових пристроїв. Управляюча частина таких симісторів працює від напруги 12В і споживає від 50мА до 100мА. (Іноземні аналоги симісторів мають ще менший струм споживання).

Коли відомий струм управління, легко знайти необхідний коефіцієнт підсилення струму управління:

$$K_n = \frac{I_y}{I_{вих}}. \quad (5.3)$$

де I_y – струм управління симістора; $I_{вих}$ – вихідний струм мікросхеми або мікропроцесора.

Вихідний струм мікросхем або мікропроцесора можна знайти в довідниках. Для елементів ТТЛШ – логіки $I_{вих} = 8$ мА. Для елементів МОП – логіки та для деяких мікропроцесорів $I_{вих} = 0,2$ мА.

Коли відомий коефіцієнт підсилення з довідників вибирають транзистор з таким коефіцієнтом підсилення по струму. Якщо такі транзистори відсутні, можна застосувати двокаскадний підсилювач.

Після розробки всіх елементів принципової схеми, розраховують потужність, яку споживає система управління. Цей розрахунок необхідний

для вибору потужності джерела живлення. Такий розрахунок не вимагає великої точності, а його результат можна округлити до найближчого більшого цілого значення.

Для мікропрограмних автоматів необхідно розробити структуру блоку пам'яті. Як говорилося раніше, блок пам'яті – це комбінаційна схема типу “один з...”. Для розробки блоку пам'яті необхідно закодувати мікрокоманди, тобто визначити, якому виходу дешифратора відповідає та чи інша мікрокоманда. Це кодування може бути виконане довільно. Структура блоку пам'яті залежить також від типу тригерів, на яких розробляється реєстр команд.

Регістри в інтегральному виконанні практично не підходять для мікропрограмних автоматів, тому що вони є синхронними елементами. Синхронізувати роботу механізмів дуже важко, тому що їх час роботи не є постійною величиною і залежить від навантаження на них. Тому краще за все реалізувати реєстр команд на асинхронних *RS*-тригерах. При виконанні мікрокоманди на виході блоку пам'яті нульовий код, а тому такі тригери не переключаються. Коли необхідно змінити код мікрокоманди, на входи тригерів з блоку пам'яті повинні поступати сигнали, які переключають тригери в відповідний стан. Момент цього переключення визначається станом датчиків, а не синхроімпульсами.

Розглянемо процедуру розробки блоку пам'яті для мікропрограмного автомата, структурна схема якого наведена на рисунку 6.2.

Мікропрограмний автомат має дві мікрокоманди, а тому його дешифратор повинен мати три виходи (один вихід використовують для виключення автомата).

Закодуємо мікрокоманди:

Виключення автомата – 00

K_1 – 01;

K_2 – 10.

Для цього мікропрограмного автомата необхідно мати реєстр команд, який складається з двох *RS*-тригерів.

– Назва “блок пам'яті” в мікропрограмному автоматі є не зовсім коректною. Таку назву можна застосовувати, якщо в нього включити реєстр команд. Ми застосовуємо цей термін лише тому, що його застосовують розробники теорії мікропрограмних автоматів, наприклад, [7]. По суті те, що називають блоком пам'яті, є комбінаційна схема типу “один з...”

Блок пам'яті зобразимо у вигляді діодної матриці “АБО” (рис. 5.6). Вихід тригера R_1 , S_1 є молодшим розрядом коду.

Такі матриці вже давно не застосовують, але їх наочність дозволяє досить просто знайти структуру блоку пам'яті.

На цьому рисунку чорними точками показані діоди, які з'єднують відповідні горизонтальні та вертикальні шини. Матриця заповнена за таким правилом:

- сигнал A повинен переключити дешифратор на виконання мікрокоманди K_2 , тобто встановити перший тригер в нульовий стан, а другий – в одиничний;
- сигнал B повинен переключити дешифратор на виконання мікрокоманди K_1 , тобто встановити перший тригер в одиничний стан, а другий – в нульовий;
- натиснення кнопки “ПУСК” повинно переключити дешифратор на виконання мікрокоманди K_1 , тобто встановити перший тригер в одиничний стан, а другий – в нульовий;
- натиснення кнопки “СТОП” повинно встановити обидва тригери в нульовий стан, що виключає МПА.

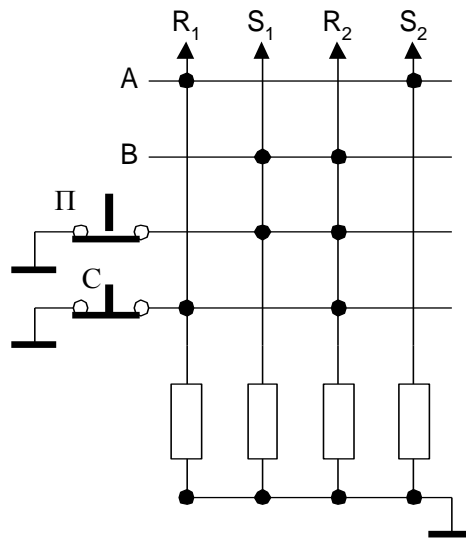


Рис. 5.6. Матриця “АБО” блоку пам’яті

З матриці можемо написати рівняння управління тригерами регістра:

$$R_1 = A + C, \quad (5.4)$$

$$S_1 = B + \Pi, \quad (5.5)$$

$$R_2 = B + \Pi + C, \quad (5.6)$$

$$S_2 = A. \quad (5.7)$$

Ці рівняння отримані за таким правилом: рівняння управління входом тригера дорівнює логічній сумі сигналів цього входу, тобто сумі сигналів, для яких є діод на відповідній горизонтальній шині.

Якщо тригери управляються одиничними сигналами, то функціональна схема блоку пам’яті має вигляд, наведений на рис. 5.7.

Якщо тригери управляються нульовими сигналами (ТТЛ-логіка), рівняння (5.4 - 5.7) необхідно інвертувати:

$$\bar{R}_1 = \overline{A + C} = \bar{A} \cdot \bar{C},$$

$$\bar{S}_1 = \overline{B + \Pi} = \bar{B} \cdot \bar{\Pi},$$

$$\bar{R}_2 = \overline{B + \Pi + C} = \bar{B} \cdot \bar{\Pi} \cdot \bar{C},$$

$$\bar{S}_2 = \bar{A}.$$

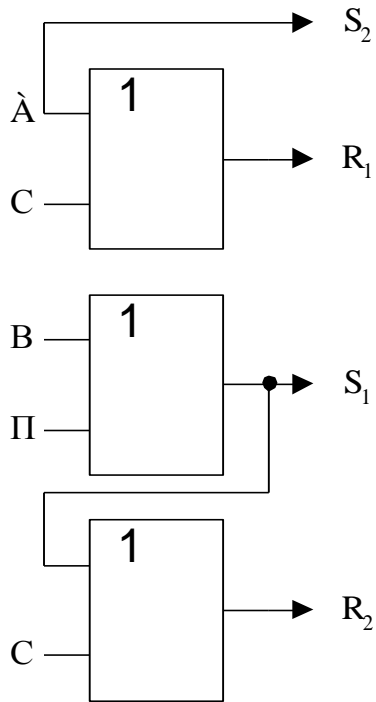


Рис. 5.7. Функціональна схема блоку пам'яті

Такі рівняння можна реалізувати або на елементах Пірса (АБО-НІ), або на елементах "І", як показано на рис. 5.8.

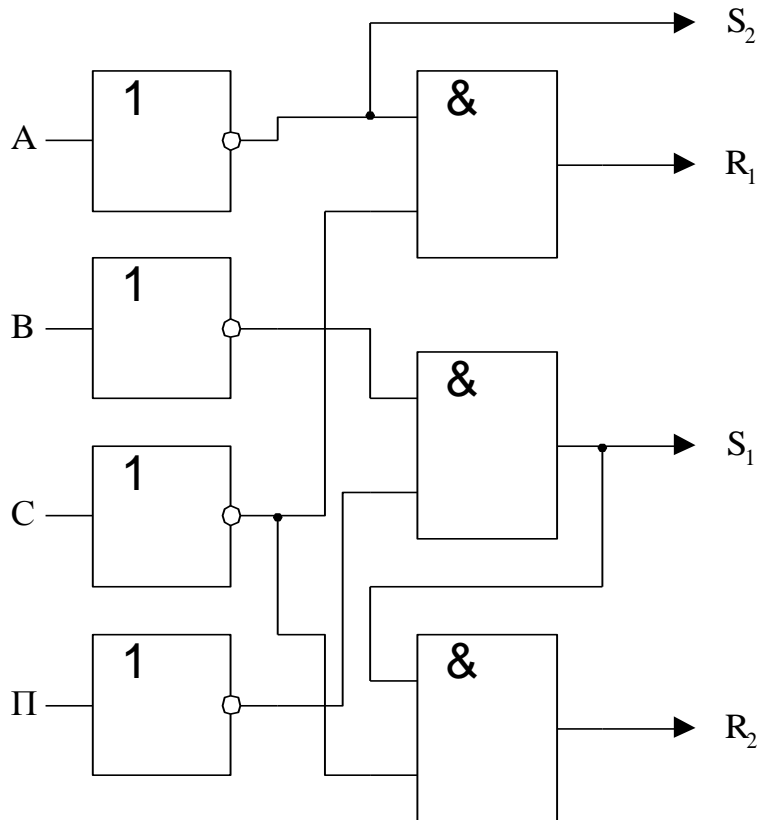


Рис. 5.8. Функціональна схема блоку пам'яті на елементах "І"

Для мікропроцесорних систем необхідно розробити програму роботи мікропроцесора. Такі програми розробляються по блок – схемі алгоритмів. Але перед цим необхідно розподілити вхідні та вихідні сигнали між відповідними вхідними та вихідними портами мікропроцесора. Для деяких типів мікропроцесорів (наприклад, МК51) кожний порт може бути вхідним або вихідним, що досягається відповідною настройкою мікропроцесора.

Для зменшення кількості входів мікропроцесора замість кнопки “ПУСК” та “СТОП” застосуємо тумблер, замкнений стан якого відповідає натиснутій кнопці “ПУСК”, а незамкнений – кнопці “СТОП”.

Визначимо порт $P1$ як вхідний і подамо на нього сигнали від датчиків та від тумблера:

- $P1.0$ – тумблер T ;
- $P1.1$ – датчик X_1 ;
- $P1.2$ – датчик X_2 ;
- $P1.3$ – датчик X_3 ;
- $P1.4$ – датчик X_4 ;
- $P1.5$ – датчик X_5 ;
- $P1.6$ – датчик X_6 ;
- $P1.7$ – датчик X_7 .

Визначимо порт $P2$ як вихідний і його виходи подамо на вхід підсилювачів:

- $P2.0$ – механізму $M1$;
- $P2.1$ – механізму $M2$;
- $P2.2$ – механізму $M3$;
- $P2.3$ – механізму $M4$;
- $P2.4$ – механізму $M5$;
- $P2.5$ – механізму $M6$.

Вихід $P2.6$ використаємо як оперативну комірку пам’яті на один біт.

За цих умов програма роботи мікропроцесора по рівняннях (2.1 - 2.14) має вигляд, наведений в таблиці 5.1.

Таблиця 5.1

Програма роботи КР1816ВЕ51

	MOV P1, # 255	настроювання порту $P1$ як вхідний;
	MOV P2, # 0	настроювання порту $P2$ як вихідний;
1:	MOV C, P1.0	визначення стану тумблера T ;
	JNC 1	
	MOV C, P2.0	Визначення стану механізму $M1$;
	JB 2	
	MOV C, P1.1	Визначення стану датчика X_1 ;
	JNC 3	
	MOV C, P1.5	Визначення стану датчика X_5 ;
	JNC 3	

	MOV C, P2.1	Визначення стану механізму M_2 ;
	JB 3	
	MOV P2.0, # 1	Включення механізму M_1 ;
	AJMP	
2:	MOV C, P1.2	Визначення стану датчика X_2 ;
	JNC 3	
	MOV P2.0, # 0	Виключення механізму M_1 ;
3:	MOV C, P2.1	Визначення стану механізму M_2 ;
	JB 4	
	MOV C, P1.2	Визначення стану датчика X_2 ;
	JNC 5	
	MOV C, P2.0	Визначення стану механізму M_2 ;
	JB 5	
	MOV P2.1, # 1	Включення механізму M_2 ;
	AJMP	
4:	MOV C, P1.3	Визначення стану датчика X_3 ;
	JNC 5	
	MOV P2.1 # 0	Виключення механізму M_2 ;
5:	MOV C, P2.6	Визначення стану тригера T ;
	JB 6	
	MOV C, P2.0	Визначення стану механізму M_1 ;
	JNC 7	
	MOV P2.6, # 1	Включення тригера T ;
	AJMP 7	
6:	MOV C, P2.3	Визначення стану механізму M_4 ;
	JNC 7	
	MOV P2.6, # 0	Виключення тригера T ;
7:	MOV C, P2.2	Визначення стану механізму M_3 ;
	JB 8	
	MOV C, P2.6	Визначення стану тригера T ;
	JNC 9	
	MOV C, P1.3	Визначення стану датчика X_3 ;
	JNC 9	
	MOV C, P2.3	Визначення стану механізму M_4 ;
	JB 9	
	MOV P2.2, # 1	Включення механізму M_4 ;
	AJMP 9	
8:	MOV C, P1.4	Визначення стану датчика X_4 ;
	JNC 9	
	MOV P2.2,# 0	
9:	MOV C, P2.3	Визначення стану механізму M_4 ;

	JB 10	
	MOV C, P1.4	Визначення стану датчика X_4 ;
	JNC 11	
	MOV C, P2.2	Визначення стану механізму $M3$;
	JB 11	
	MOV P2.3, # 1	Включення механізму $M4$;
	AJMP 11	
10	MOV C, P1.5	Визначення стану датчика X_5 ;
	JNC 11	
	MOV P2.3, # 0	Виключення механізму $M4$;
11	MOV C, P2.4	Визначення стану механізму $M5$;
	JB 12	
	MOV C, P2.0	Визначення стану механізму $M1$;
	JNC 13	
	MOV C, P2.5	Визначення стану механізму $M6$;
	JB 13	
	MOV P2.4, # 1	Включення механізму $M5$
	AJMP 13	
12	MOV C, P1.6	Визначення стану датчика X_6 ;
	JNC 13	
	MOV P2.4, # 0	Виключення механізму $M5$;
13	MOV C, P2.5	Визначення стану механізму $M6$;
	JB 14	
	MOV C, P1.6	Визначення стану датчика X_6 ;
	JNC 1	
	MOV C, P2.4	Визначення стану механізму $M5$;
	JB 1	
	MOV P2.5, # 1	Включення механізму $M6$;
	AJMP 1	
14	MOV C, P1.7	Визначення стану датчика X_7 ;
	JNC 1	
	MOV P2.5, # 0	Виключення механізму $M6$;
	AJMP 1	

Подібним чином складається програма роботи мікропроцесора по ЛСА. Програма роботи мікропроцесора по блок – схемі алгоритмів 5.5 наведена в табл. 5.2. Розподіл входів і виходів мікропроцесора проведемо так, як це було зроблено раніше.

Таблиця 5.2

Програма роботи КР1816ВЕ51 згідно ЛСА (5.2)

	MOV P1, # 255	Настроювання порту P1 як вхідний;
	MOV P2, # 0	Настоявання порту P2 як вихідний;
1:	MOV C, P1.0	Визначення стану тумблера T;
	JNC 1	
2:	MOV C, P1.1	Визначення стану датчика X ₁ ;
	JNC 2	
	MOV P2.0, # 1	Включення механізму M1;
3:	MOV C, P1.2	Визначення стану датчика X ₂ ;
	JNC 3	
	MOV P2.0, # 0	Виключення механізму M1;
	MOV P2.1, # 1	Включення механізму M2:
4:	MOV c, P1.3	Визначення стану датчика X ₃ ;
	JNC 4	
	MOV P2.1, # 0	Виключення механізму M2;
	MOV P2.2, # 1	Включення механізму M3;
5:	MOV C, P1.4	Визначення стану датчика X ₄ ;
	JNC 5	
	MOV P2.2, # 0	Виключення механізму M3;
	MOV P2.3, # 1	Включення механізму M4;
6:	MOV C, P1.5	Визначення стану датчика X ₅ ;
	JNC 6	Виключення механізму M4;
	MOV P2.3, # 0	
	MOV P2.4, # 1	Включення механізму M5;
7:	MOV C, P1.6	Визначення стану датчика X ₆ ;
	JNC 7	
	MOV P2.4, # 0	Виключення механізму M5;
	MOV P2.5, # 1	
8:	MOV C, P1.7	Визначення стану датчика X ₇ ;
	JNC 8	
	MOV P2.5, # 0	Виключення механізму M6;
	AJMP 1	

5.3. Контрольні питання до розділу 5

1. Які основні елементи входять до складу структурної схеми системи управління, що побудована на цифрових елементах?
2. Поясніть реалізацію мікрокоманд у схемі мікропрограмного автомату (МПА) на дешифраторах та логічних елементах.
3. Дайте характеристику структурної схеми мікропроцесорної системи управління.
4. Як розробити алгоритм роботи системи управління на основі рівнянь управління?
5. Як розробити алгоритм роботи системи управління на основі логічної схеми алгоритму (ЛСА)?
6. Навіщо до структурної схеми системи управління додаються кнопки «ПУСК» та «СТОП»?
7. Принципи вибору елементної бази для електричної принципової схеми системи управління.
8. Як розробити функціональну схему блоку пам'яті МПА?
9. Що потрібно підключити до вхідного та вихідного портів мікропроцесорної системи управління?

РОЗДІЛ 6

ПРОЕКТУВАННЯ СИСТЕМ АВТОМАТИЧНОГО РЕГУЛЮВАННЯ ТА КОНТРОЛЮ ПАРАМЕТРІВ ОБ'ЄКТІВ

6.1. Загальна характеристика систем автоматичного регулювання

Принципи проектування пристроїв та систем управління, які викладені в попередніх розділах, можна застосовувати до об'єктів, в яких датчики та виконавчі механізми знаходяться лише в двох станах — ввімкнено/вимкнено. Це логічні об'єкти, закони управління якими базуються на математичній логіці.

Не менш поширене застосування знаходять об'єкти, в яких необхідно підтримувати на заданому рівні або змінювати по заданому закону певний параметр, який характеризує технологічний процес в цьому об'єкті. Такі системи називають системами автоматичного регулювання (САР). Як правило, між вхідними та вихідними сигналами таких систем існує певний безперервний зв'язок. При цьому самі сигнали можуть бути представлені як в аналоговій так і цифровій формі. Основою таких систем є теорія автоматичного регулювання.

Історично теорія автоматичного регулювання виникла раніше теорії логічних систем. САР в наш час має не тільки розвинену теорію, але й досить потужну матеріальну базу у вигляді датчиків різноманітних параметрів, регуляторів та виконавчих механізмів, випуск яких обчислюється сотнями тисяч примірників на рік і які входять в державну систему приладів.

У більшості випадків цих приладів та регуляторів достатньо для розробки практично будь-якої системи регулювання.

Формулювання робочого завдання для САР найчастіше виконує оператор по тим чи іншим ознакам технологічного процесу. Також намітилися тенденція включати регулятори до складу системи управління верхнього рівня, яка і формулює це завдання.

Проектування САР починають з вивчення об'єкта регулювання. Необхідно в'яснити всі вхідні та вихідні параметри об'єкта регулювання, а також функціональні залежності між ними, межу зміни параметрів і необхідну точність регулювання. Необхідно вибрати датчики параметрів і місце їх розташування на об'єкті. По зв'язкам між вхідними та вихідними величинами визначаються статична та динамічна модель об'єкта і необхідний закон регулювання.

Дуже часто вибір закону регулювання та тип необхідної апаратури виконується на основі порівняння даного об'єкта з аналогічним об'єктом.

При необхідності властивості об'єкта визначаються шляхом експерименту.

При проектування САР необхідно розробляти ту ж саму технічну документацію, що й при проектуванні логічних систем управління.

6.2. Основні терміни та визначення

В теорії автоматичного регулювання використовується термінологія, яка склалася на протязі декількох століть. Найважливішими з них є:

1. Об'єкт автоматичного регулювання або просто об'єкт — технологічний агрегат, в якому необхідно підтримувати на заданому рівні або змінювати по заданому закону певний параметр (нагрівальна піч, котел, турбіна тощо).

2. Регулюємий параметр або просто параметр — величина, яка визначає якість технологічного процесу (температура, тиск, вага, об'єм, рівень тощо).

3. Регулююча величина (вхідна величина) — речовина або енергія, зміна якої призводить до зміни параметру.

В кожному об'єкті має місце перетворення речовини або енергії. Тому в об'єкті завжди щось подають ($Q_{\text{под}}$) і щось з нього виходить. Частина речовини або енергії витрачається в об'єкті. Це витрати або навантаження ($Q_{\text{вит}}$).

Технологічний процес характеризується матеріальним або (та) енергетичним балансом. Тому, якщо $Q_{\text{под}} = Q_{\text{вит}}$, то параметр не змінюється, $\frac{d\sigma}{dt} = 0$. Якщо $Q_{\text{под}} > Q_{\text{вит}}$, то параметр збільшується, $\frac{d\sigma}{dt} > 0$. Якщо $Q_{\text{под}} < Q_{\text{вит}}$,

то параметр зменшується, $\frac{d\sigma}{dt} < 0$.

Система регулювання діє або на $Q_{\text{под}}$ (найчастіше) або на $Q_{\text{вит}}$, що й призводить до зміни параметру в необхідному напрямку.

На величини $Q_{\text{под}}$ та $Q_{\text{вит}}$ можуть впливати і інші фактори, які називаються збуреннями і які впливають на величину параметру. Так, якщо відкрити дверці нагрівальної печі, то температура в ній знизиться через збільшення витрат тепла.

Частина збурень може бути відома і їх вплив можна урахувати. Найчастіше більшість збурень невідома. Але через те, що їх вплив проявляється в зміні параметру, його може ліквідувати система регулювання.

Один і той же об'єкт може мати декілька параметрів. Наприклад, температура в нагрівальній печі визначається тепловим балансом і залежить від подачі газу та повітря. Тиск в цій же печі визначається матеріальним балансом газу, повітря та продуктів згорання. В таких об'єктах зміна одного параметру може призвести до зміни іншого, що необхідно враховувати при розробці САР.

Якщо зміна параметру в об'єкті призводить до зменшення величини $\Delta Q = Q_{\text{под}} - Q_{\text{вит}}$, то такий об'єкт називається статичним або об'єктом з самовирівнюванням. Ці властивості об'єкта допомагають регулятору ліквідувати відхилення параметру від заданого значення. Прикладом такого об'єкту може бути ємність, в якій збільшення рівня рідини збільшує її тиск,

що призводить до збільшення кількості рідини, яка виходить з ємності через отвір в її нижній частині. Тому остаточна зміна рівня рідини в ємності буде меншою.

Якщо зміна параметру не змінює величини ΔQ , то такий об'єкт називається астатичним або об'єктом без самовирівнювання. Для таких об'єктів потрібні більш складні САР. Прикладом такого об'єкту може бути та ж ємність, з якої рідина відбирається насосом, продуктивність якого практично не залежить від рівня рідини в ємності.

Найважливішими характеристиками об'єкта є такі:

1. Час запізнення ($\tau_{\text{зап}}$) — час від початку дії до початку зміни параметру. Запізнення негативно впливає на процес регулювання, тому що сприяє виникненню коливань параметру.

2. Постійна часу (T) визначає швидкість зміни параметру в об'єкті при порушенні рівноваги між $Q_{\text{под}}$ та $Q_{\text{вит}}$. Чим менша величина T , тим більша швидкість зміни параметру. З іншого боку такий об'єкт в більшій мірі реагує на дію регулятора. Тому для САР більш важливим є співвідношення $\tau_{\text{зап}}/T$.

3. Коефіцієнт передачі (коефіцієнт підсилення) — зміна параметру при одиничному збуренні:

$$K_{\text{об}} = \frac{X_y - X_0}{\Delta Y},$$

де X_y — нове значення параметру;

X_0 — початкове значення параметру;

ΔY — переміщення регулюючого органу або зміна ΔQ .

Чим більше значення $K_{\text{об}}$, тим в більшій мірі впливає регулююча дія на зміну параметру, але і тим більша вірогідність виникнення його коливань.

Ці характеристики об'єкта необхідні при розробці САР і визначаються теоретичним або експериментальним шляхом.

Один і той же об'єкт по різним параметрам має різні характеристики.

САР можуть бути побудовані за двома основними принципами:

1. Принцип відхилення.

2. Принцип збурення.

Структурна схема САР першого типу є замкнутою і наведена на рис. 6.1. В такій системі регулятор змінює $Q_{\text{п}}$ (можливо змінювати $Q_{\text{вит}}$) в тому випадку, коли є різниця між дійсним $\sigma_{\text{д}}$ та заданим $\sigma_{\text{з}}$ значенням параметру. Ця дія регулятора можлива лише тоді, коли $\Delta\sigma = \sigma_{\text{д}} - \sigma_{\text{з}} \neq 0$. Тому в такій системі параметр деякий час може не дорівнювати заданому. Але в такій САР автоматично враховуються всі збурення — відомі та невідомі.

В САР другого типу дія регулятора має місце при появі збурення. Це дозволяє не допустити відхилення параметру від заданого значення. В такій системі немає зворотного зв'язку по параметру, тому останній може і не дорівнювати заданому при появі невідомих збурень. Через це САР такого типу, як самостійні системи, не застосовуються.

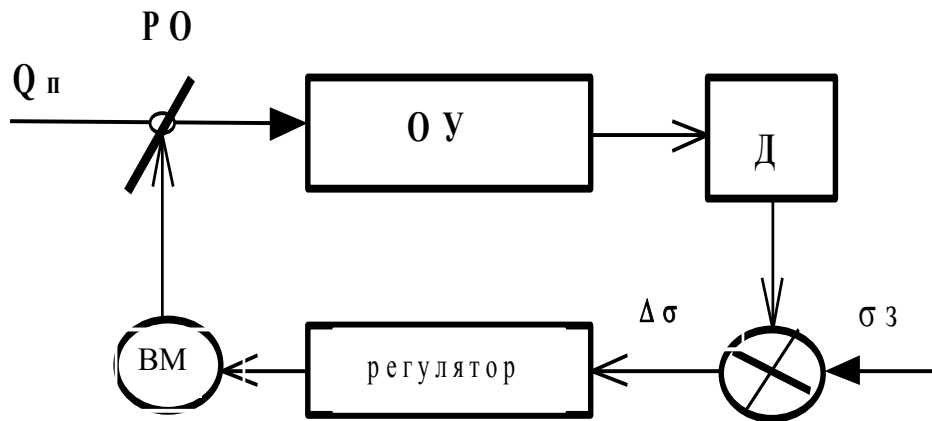


Рис. 6.1

6.3. Вибір регулятора

Регулятори можуть бути безперервної дії, релейні та імпульсні. В регуляторах безперервної дії між відхиленням параметру ($\Delta\sigma$) та переміщенням виконавчого механізму (або регулюючого органа) є безперервний функціональний зв'язок.

В релейних регуляторах виконавчий механізм включається, якщо $\Delta\sigma > x$, причому, як правило, швидкість його руху є постійною величиною, а напрям руху визначається знаком відхилення.

В імпульсних регуляторах управляючий сигнал являє собою послідовність імпульсів, частота, тривалість або амплітуда яких залежить від $\Delta\sigma$.

Регулятори можуть бути прямої або непрямої дії. Останні мають в своєму складі підсилювачі.

По типу енергії регулятори можуть бути електричні, пневматичні, гідравлічні або комбіновані.

Але найбільш суттєвою є класифікація регуляторів по виду зв'язку між вхідним сигналом ($\Delta\sigma$) та переміщенням виконавчого механізму (регулюючого органа). Формування цих зв'язків здійснюється або за рахунок зворотніх зв'язків, або за рахунок спеціальних елементів, або за рахунок і перших і других.

Закони регулювання виконуються з певними обмеженнями в області нормальних режимів роботи регулятора. Ці режими залежать від динамічних властивостей як регулятора так і об'єкта. Тому кожний регулятор необхідно наставляти, тобто його властивості необхідно узгодити з властивостями об'єкта. Для цього регулятори мають відповідні елементи.

Для вибору відповідного типу регулятора необхідно познайомитися з їх властивостями. Найбільш поширеними є чотири типи регуляторів.

Астатичні або інтегруючі регулятори (І-регулятори). В таких регуляторах зв'язок між відхиленням параметру та переміщенням виконавчого механізму має вигляд:

$$\frac{d\mu}{dt} = K_p \cdot \Delta\sigma \text{ або } \mu = K_p \cdot \int \Delta\sigma \cdot dt$$

де μ — переміщення виконавчого механізму; K_p — коефіцієнт підсилення регулятора; $\Delta\sigma$ — відхилення параметру від заданого значення.

В таких регуляторах виконавчий механізм не включається лише тоді, коли $\Delta\sigma = 0$, тобто він завжди прагне ліквідувати відхилення параметру, незалежно від причини його виникнення. Ці позитивні властивості регулятора призводять до виникнення коливань в системі.

Нехай маємо астатичний об'єкт без запізнення та І-регулятор без зони нечутливості (без величини $\Delta\sigma$, на яку ще не реагує регулятор). Перехідний процес в такій САР зображений на рис. 6.2.

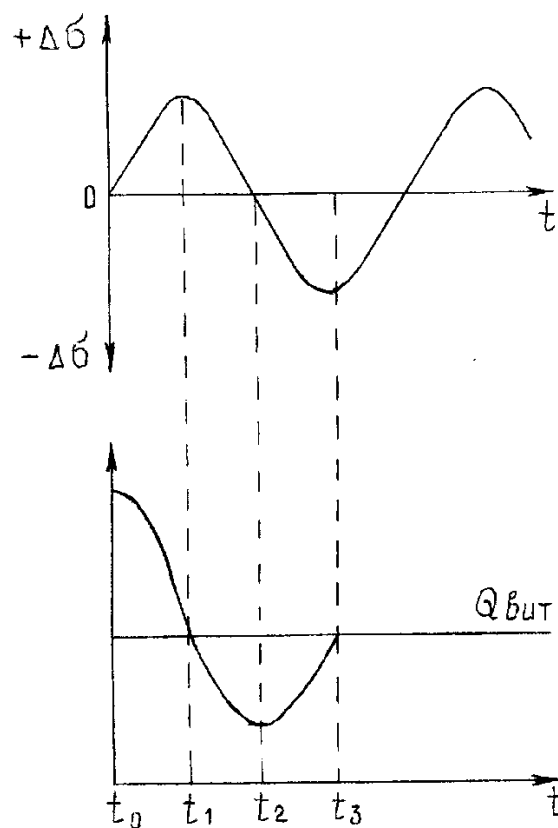


Рис. 6.2

Нехай до моменту часу $t = t_0$, $Q_{\text{под}} = Q_{\text{внт}}$, а $\Delta\sigma = 0$. В момент часу $t = t_0$ $Q_{\text{под}}$ стало меншим за $Q_{\text{внт}}$ і параметр в об'єкті починає збільшуватися зі швидкістю, яка визначається його постійною часу. При $\Delta\sigma > 0$, регулятор починає переміщувати регулюючий орган, що зменшує величину $Q_{\text{под}}$. В момент часу $t = t_1$ $Q_{\text{под}} = Q_{\text{внт}}$ і $\frac{d\sigma}{dt} = 0$, тобто параметр більше не змінюється.

Але $\Delta\sigma \neq 0$, тому регулятор продовжує переміщення регулюючого органу в тому ж напрямку і $Q_{\text{под}}$ стає меншою за $Q_{\text{внт}}$. Це призводить до того, що параметр починає зменшуватися. В момент часу $t = t_2$ $\Delta\sigma = 0$ і виконавчий

механізм зупиняється. Але $Q_{\text{под}} < Q_{\text{вит}}$, тому параметр продовжує зменшуватися і його значення стає меншим за задане. Це призводить до нового включення виконавчого механізму, але в іншому напрямку.

В системі виникають безперервні коливання параметру відносно σ_3 і $Q_{\text{под}}$ відносно $Q_{\text{вит}}$.

При наявності запізнення в об'єкті або при наявності зони нечутливості у регулятора амплітуда коливань параметру буде збільшуватися і система САР стає нестійкою. Тому на астатичних об'єктах астатичні регулятори не встановлюють.

Якщо об'єкт статичний, то зміна параметру викликає зміну $Q_{\text{вит}}$. В цьому разі рівновага між $Q_{\text{под}}$ та $Q_{\text{вит}}$ встановлюється скоріше і це зменшує максимальне відхилення параметру. В такій системі коливання будуть затухати і параметр буде повернуто в початковий стан (рис. 6.3).

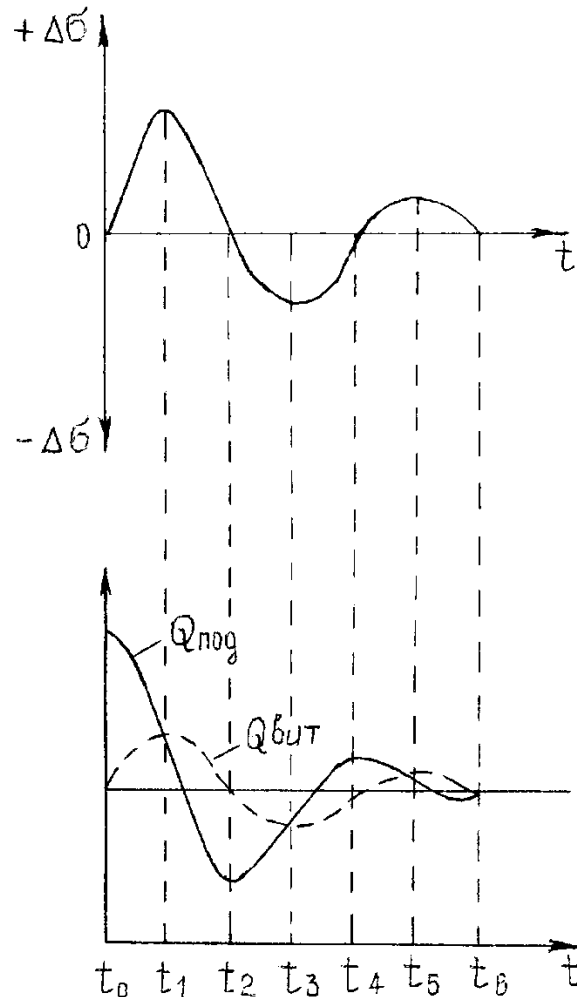


Рис. 6.3

В практичних САР дуже часто швидкість переміщення виконавчого механізму є величиною постійною. Яка б не була малою ця швидкість, все ж таки є такі відхилення, для яких вона буде надмірною. В таких САР параметр може колитися в межах, які є досить малі і які з технологічної точки зору не мають значення. Але виконавчий механізм весь час включається, що

негативно впливає на його працездатність. В цьому випадку регулятор повинен мати зону нечутливості, щоб не реагувати на такі малі відхилення.

Пропорційні або статичні регулятори (П-регулятори).

В таких регуляторах зв'язок між відхиленням параметру та переміщенням виконавчого механізму має вигляд:

$$\frac{d\mu}{dt} = K_p \cdot \frac{d\sigma}{dt} \quad \text{або} \quad \mu = K_p \cdot \Delta\sigma.$$

В таких регуляторах виконавчий механізм переміщається лише тоді, коли є зміна параметру, тобто тоді, коли $Q_{\text{под}} \neq Q_{\text{вит}}$. Перехідний процес в таких регуляторах зображений на рис. 6.4.

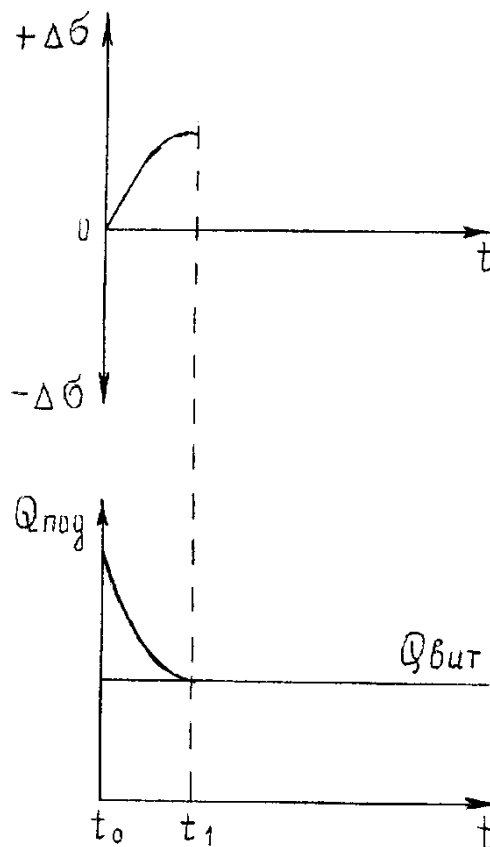


Рис. 6.4

При виникненні умови $Q_{\text{под}} > Q_{\text{вит}}$ параметр починає збільшуватися і регулятор починає переміщувати регулюючий орган, який зменшує $Q_{\text{под}}$, що зменшує і швидкість зміни параметру. В момент часу $t = t_1$ встановлюється рівновага між подачею і витратами і параметр перестає змінюватися. Регулятор виключає виконавчий механізм і на цьому перехідний процес закінчується.

Таким чином пропорційний регулятор досить швидко встановлює рівновагу в об'єкті, тобто гасить коливання в об'єкті. Але при цьому параметр може і не бути рівним заданому, що серйозним недоліком таких регуляторів. Відхилення $\Delta\sigma_{\text{зал}}$ є залишковим і називається статичною похибкою. Через те пропорційні регулятори практично не застосовуються.

Пропорційно-інтегруючі або ізодромні регулятори (ПІ-регулятори).

В таких регуляторах зв'язок між відхиленням параметру і переміщенням виконавчого механізму має такий вигляд:

$$\frac{d\mu}{dt} = K_1 \cdot \Delta\sigma + K_2 \cdot \frac{d\sigma}{dt} \quad \text{або} \quad \mu = K_1 \cdot \int \Delta\sigma \cdot dt + K_2 \cdot \Delta\sigma.$$

Такі регулятори поєднують в собі властивості І- та П-регуляторів.

При порушенні рівноваги між подачею і витратами параметр починає змінюватися, причому на початку цього процесу швидкість зміни параметру велика, а величина відхилення ще мала. Тому в основному працює пропорційна частина регулятора, яка досить швидко встановлює рівновагу між подачею і витратами. В цей момент часу швидкість зміни параметру практично дорівнює нулю, а відхилення має максимальне значення.

Тому в основному працює інтегральна частина регулятора, яка змінює подачу таким чином, щоб параметр почав повертатися в початковий стан.

Це призводить до зміни параметру і включення пропорційної частини, яка не допускає виникнення коливань в системі (рис. 6.5). Ступінь впливу кожної з частин регулятора визначається величиною коефіцієнта K_1 та K_2 і це визначає форму перехідного процесу, яка може бути коливною або аперіодичною.

Якщо в об'єкті часто виникають збурення астатична частина такого регулятора практично не встигає включатися. В цьому разі мають місце статичні похибки, які можуть бути усунені наступним типом регулятора.

Пропорційно-інтегро-диференційні регулятори або регулятори з упередженням (ПІД-регулятори).

В таких регуляторах зв'язок між відхиленням параметру та переміщенням виконавчого механізму має такий вигляд:

$$\frac{d\mu}{dt} = K_1 \cdot \Delta\sigma + K_2 \cdot \frac{d\sigma}{dt} + K_3 \cdot \frac{d^2\sigma}{dt^2}.$$

Такі регулятори реагують на відхилення параметру (астатична частина), на швидкість відхилення (статична частина) та на прискорення параметру (упереджувальна частина). Тому при одних і тих же умовах якість регулювання у них буде кращою ніж у інших типах регуляторів.

Ускладнення закону регулювання викликає і ускладнення схемних рішень і, відповідно, ціни регулятора.

Для отримання певного перехідного процесу, який визначається вимогами до САР з боку технології, необхідно вибрати відповідний тип регулятора. В об'єкті можуть бути такі типи перехідного процесу [14, 24, 291]:

- стійкий коливний перехідний процес без статичної похибки (рис. 6.6);
- стійкий коливний перехідний процес з статичною похибкою (рис. 6.7).
- аперіодичний перехідний процес без статичної похибки (рис. 6.8);
- аперіодичний перехідний процес з статичною похибкою (рис. 6.9).

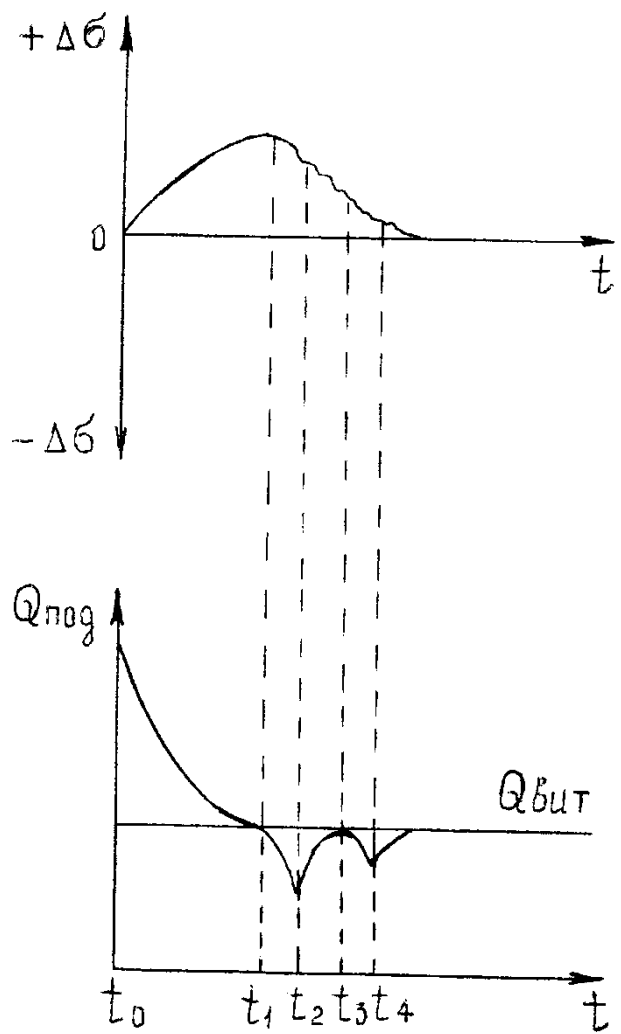


Рис. 6.5

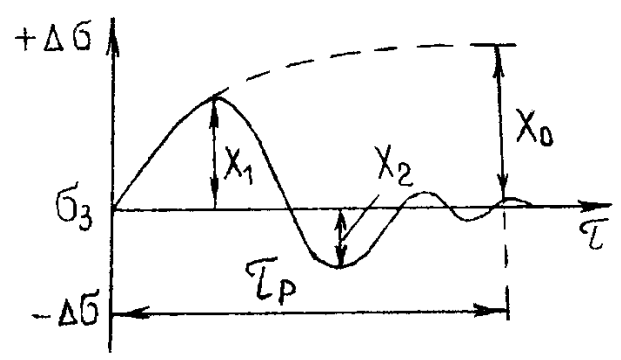


Рис. 6.6

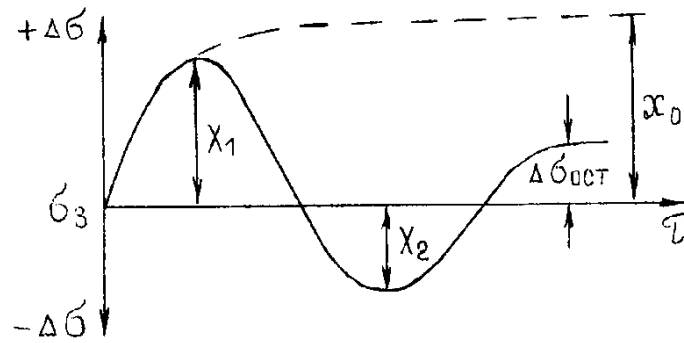


Рис. 6.7

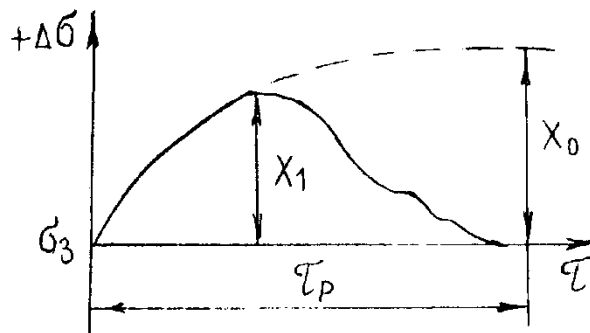


Рис. 6.8

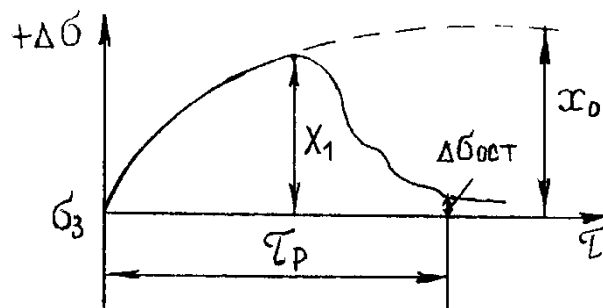


Рис. 6.9

Ступінь впливу регулятора визначається динамічним коефіцієнтом регулювання:

$$R_g = \frac{X_1}{X_0} = \frac{X_1}{K_0 \cdot \mu_0},$$

де X_1 — допустиме відхилення параметру в перехідному процесі (вибіг параметру або перерегулювання); K_0 — коефіцієнт підсилення об'єкта; μ_0 — величина збурення.

Величину X_0 можна визначити по кривій розгону (зміна параметру в часі при зміні подачі в об'єкт при відсутності регулятора), по якій визначають динамічні характеристики об'єкта.

Тип регулятора визначають по номограмам. Такі номограми створені для кожного типу перехідних процесів. Одна з номограм для аперіодичного перехідного процесу наведена на рис. 6.10. Точка на номограмі визначається необхідним значенням R_g та відношенням часу запізнення до постійної часу об'єкта. Якщо ця точка знаходиться вище кривої 1, то заданий перехідний процес можна отримати за допомогою I - регулятора. Область між кривими 1 та 2 вимагає II - регулятора. Область між кривими 2 та 3 вимагає III - регулятора, а область між кривими 3 та 4 — ПД - регулятора. Якщо точка знаходиться нижче кривої 4, то отримати такий перехідний процес на даному об'єкті неможливо.

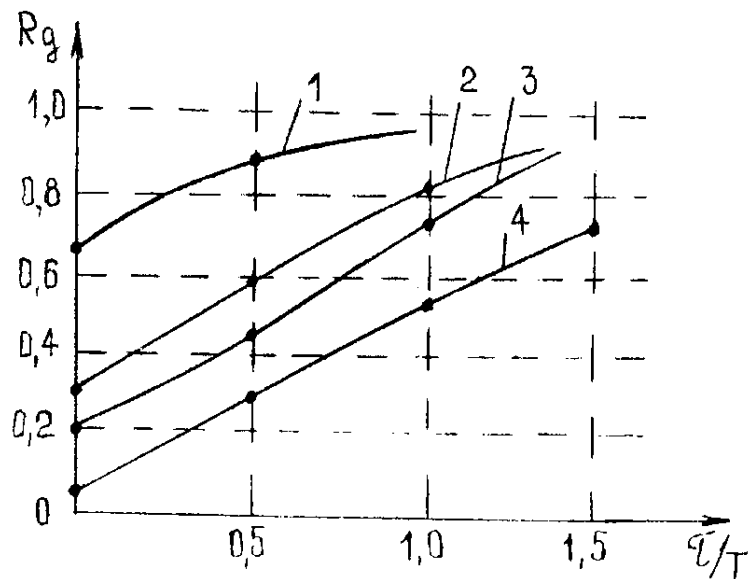


Рис. 6.10

Після вибору закону регулювання необхідно перевірити (якщо для технології це важливо) чи забезпечує вибраний регулятор необхідний час регулювання. Ця перевірка здійснюється по графікам, які розроблені для кожного типу перехідного процесу. Один з таких графіків для аперіодичного перехідного процесу наведений на рис. 6.11.

Після цього необхідно визначити статичну похибку (якщо це має значення для даного технологічного процесу). Статичну похибку можна визначити по графікам, які розроблені для кожного типу перехідного процесу. Один з таких графіків для аперіодичного перехідного процесу наведений на рис. 6.12.

Після остаточного вибору закону регулювання необхідно звернутися до довідників і вибрати конкретний тип регулятора з тих, які випускаються промисловістю. В цих довідниках вказується з якими датчиками та виконавчими механізмами може працювати даний регулятор. Необхідно впевнитися в тому, що потужність виконавчого механізму достатня для переміщення регулюючого органу.

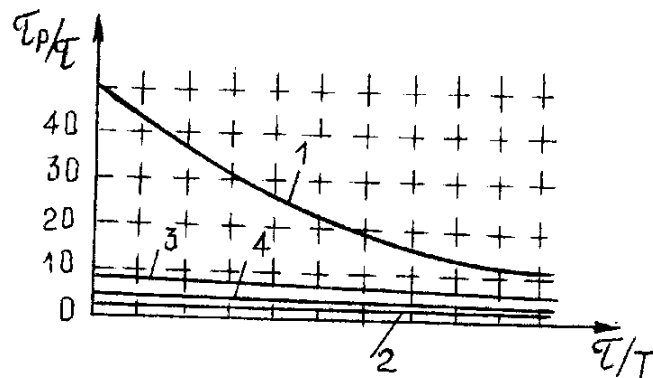


Рис. 6.11

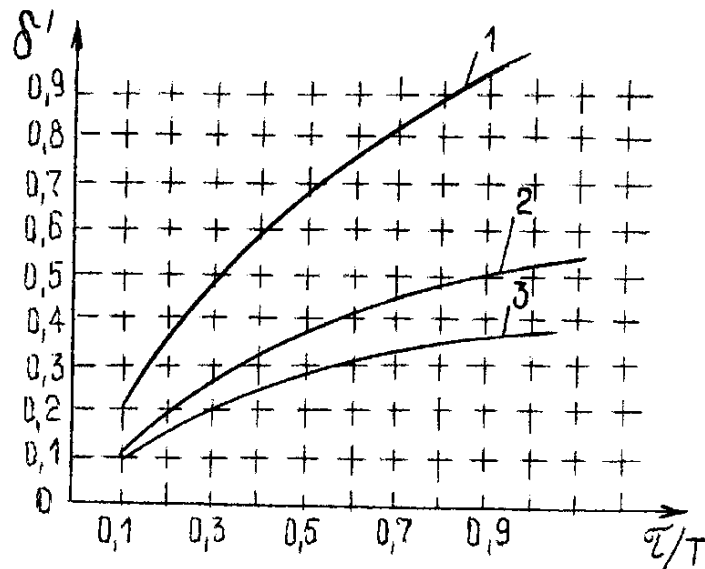


Рис. 6.12

6.4. Вибір датчика

Точність та надійність роботи САР значною мірою визначається правильним вибором датчика даного технологічного процесу.

Вибір датчика проводиться по каталогам або по довідниках, виходячи з таких вимог:

- межа зміни параметра;
- точність вимірювання;
- чутливість датчика;
- умови експлуатації датчика, тобто параметри середовища, в якому він знаходиться;
- економічні вимоги.

Верхня межа параметру повинна бути меншою за те значення, яке може сприймати датчик, що гарантує неможливість пошкодження останнього

при випадковому підвищенні параметра в об'єкті. Так при температурах більших ніж 1300°C забороняється застосовувати термопари ХК та ХА, тому що максимальне значення температури, яке може сприйматися такими термопарами 1000°C . В цьому разі необхідно застосовувати термопари ПП або ПР 30/6, хоч вони мають більшу вартість і меншу чутливість.

При температурах до 700°C більш доцільно застосовувати термометри опору, які мають більшу чутливість.

Якщо по умовам технологічного процесу необхідна реєстрація параметра, то необхідно вибрати і відповідний вимірювальний прилад. Як правило, в довідниках вказується з якими приладами може працювати даний датчик. При цьому точність вимірювання визначається точністю датчика та цього приладу.

Для багатьох датчиків розроблені спеціальні перетворювачі, які перетворюють сигнал датчика в стандартний сигнал або в вигляді струму $0 - 5\text{ mA}$, $0 - 10\text{ mA}$, або в вигляді напруги $0 - 10\text{ V}$, або в вигляді частоти $4 - 8\text{ кГц}$.

6.5. Вибір регулюючого органу

Для нормальної роботи САР необхідно правильно вибрати не тільки датчик та регулятор, але й регулюючий орган.

Призначення регулюючого органу в тому, щоб змінювати величину подачі (витрат) в об'єкті за рахунок зміни опору (електричного або гідравлічного) в системі подачі енергії або речовини.

Як регулюючі органи застосовують клапани, заслінки, шибера (прямокутна заслінка з лінійним переміщенням), а в електричних ланцюгах реостати, тиристори тощо.

Заводською характеристикою регулюючого органу є залежність між площею прохідного отвору його та переміщенням:

$$S = f(\mu)$$

Для реостата:

$$R = f(\mu)$$

Ця залежність визначається конструкцією регулюючого органу і може бути лінійною або нелінійною.

Для САР більш важливою є робоча характеристика регулюючого органу, яка визначається залежністю між переміщенням та величиною подачі:

$$Q_{np} = f(\mu) \quad \text{або} \quad Q_{вум} = f(\mu)$$

Вигляд цієї залежності визначається співвідношенням між опором регулюючого органу та опором всього тракту подачі. Тому один і той же регулюючий орган на різних об'єктах може мати різну робочу характеристику.

Нехай маємо електричне коло, що зображене на рис. 6.13.

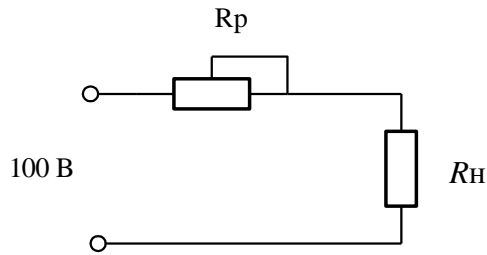


Рис. 6.13

Нехай $U = 100\text{В}$, $R_p = 100\ \text{Ом}$, $R_n = 100\ \text{Ом}$. Максимальний струм в ланцюгу дорівнює 1А , а мінімальний дорівнює $0,5\ \text{А}$. Межа регулювання струму дорівнює 2 , що замало. При $R_p = 300$, межа регулювання буде складати 4 , а при $R_p = 1000$, межа регулювання буде дорівнювати приблизно 100 .

Якщо опір регулюючого органу складає не більше 30% загального опору в системі подачі об'єкті, то його робоча характеристика має вигляд, наведений на рис. 6.14.

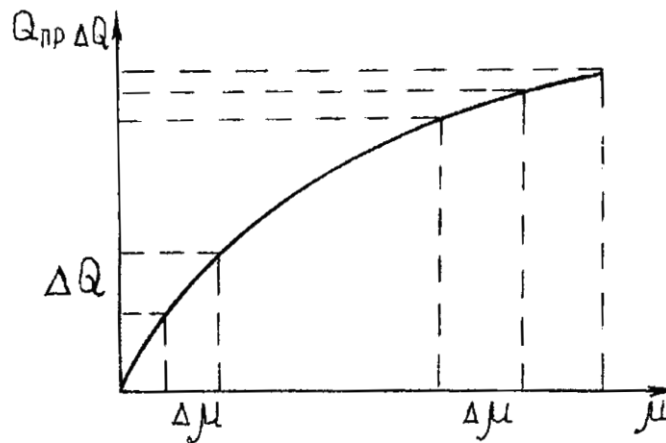


Рис. 6.14

В САР величина $\Delta \mu$ пропорційна відхиленню параметра $\Delta \sigma$. При малих навантаженнях (тобто при малих $Q_{\text{вит}}$) величина $Q_{\text{под}}$ теж повинна бути малою, тому що задача САР зводиться до підтримки рівноваги між подачею та витратами. При такій характеристиці регулюючого органу невеликі зміни його положення при малих витратах призводять до значної зміни подачі, що буде визивати великі відхилення параметру і до створення коливань в системі.

При великих навантаженнях такі ж самі відхилення параметру будуть визивати малі зміни подачі, а тому і малий вплив на параметр.

Звідси випливає, що при такій характеристиці регулюючого органу САР зі своєю задачею не справиться.

При опорі регулюючого органу, який приблизно дорівнює опору в системі подачі, його робоча характеристика буде лінійною або близькою до лінійної (рис. 6.15).

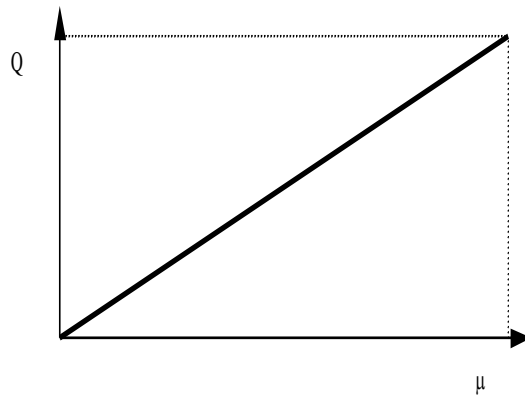


Рис. 6.15

Такий регулюючий орган однаково реагує на відхилення параметру при великих і при малих навантаженнях. Така система не зможе забезпечити необхідну якість регулювання або в області малих навантажень, або в області великих.

Найкращою характеристикою регулюючого органу буде характеристика, яка має місце тоді, коли його опір набагато (на один порядок) вище опору тракту подачі. Характеристика такого робочого органу наведена на рис. 6.16.

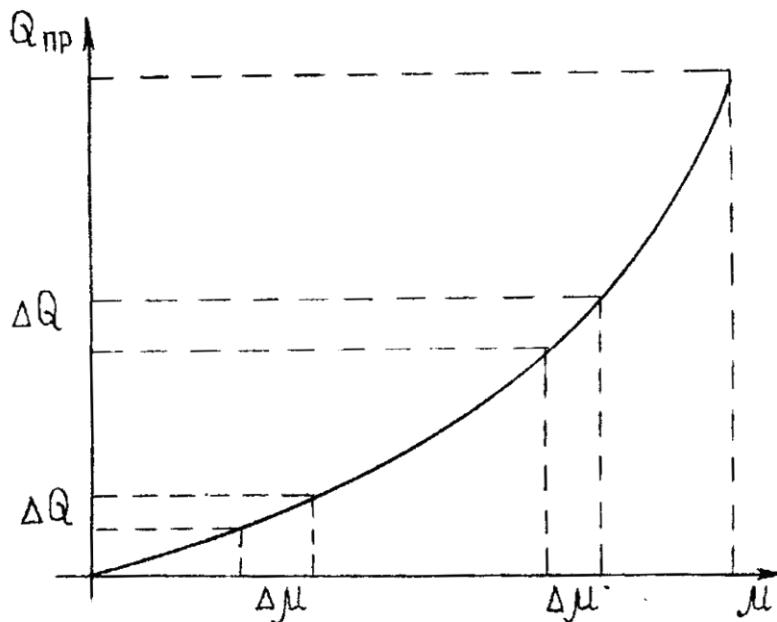


Рис. 6.16

При такій характеристиці однакові переміщення робочого органу будуть викликати малі зміни подачі при малих навантаженнях і великі – при великих, що потрібно для якісного регулювання.

6.6. Позиційні регулятори

Серед промислових регуляторів широке застосування знаходять релейні регулятори, в яких регулюючий орган може займати декілька

фіксованих положень або позицій. Найчастіше застосовуються регулятори, в яких регулюючий орган займає дві позиції, одне з яких можна назвати “ВІДКРИТО”, інше – “ЗАКРИТО”. Застосування подібних регуляторів вимагає певних співвідношень в САР. Коли регулюючий орган займає положення “ВІДК” в об’єкт поступає максимальна кількість енергії або речовини. Якщо буде витримано співвідношення

$$Q_{под}^{мак} > Q_{вит}^{мак}$$

то при положенні регулюючого органу “ВІДК” параметр завжди буде збільшуватися.

Коли регулюючий орган займає положення “ЗАКР”, в об’єкт поступає мінімальна кількість енергії або речовини. Якщо буде витримано співвідношення

$$Q_{под}^{мін} < Q_{вит}^{мін}$$

то при положенні регулюючого органу “ЗАКР” параметр завжди буде зменшуватися.

Коли такі умови будуть виконані параметр в САР здійснює незатухаючі коливання навколо заданого значення, що припустимо для багатьох видів технологічних процесів, якщо амплітуда цих коливань не буде перевищувати допустимих значень.

Реальні регулятори мають зону нечутливості, в межах якої регулятор не реагує на відхилення параметру. Ширина цієї зони нечутливості є елементом настроювання регулятора.

Процес регулювання в такій САР здійснюється таким чином (рис. 6.17).

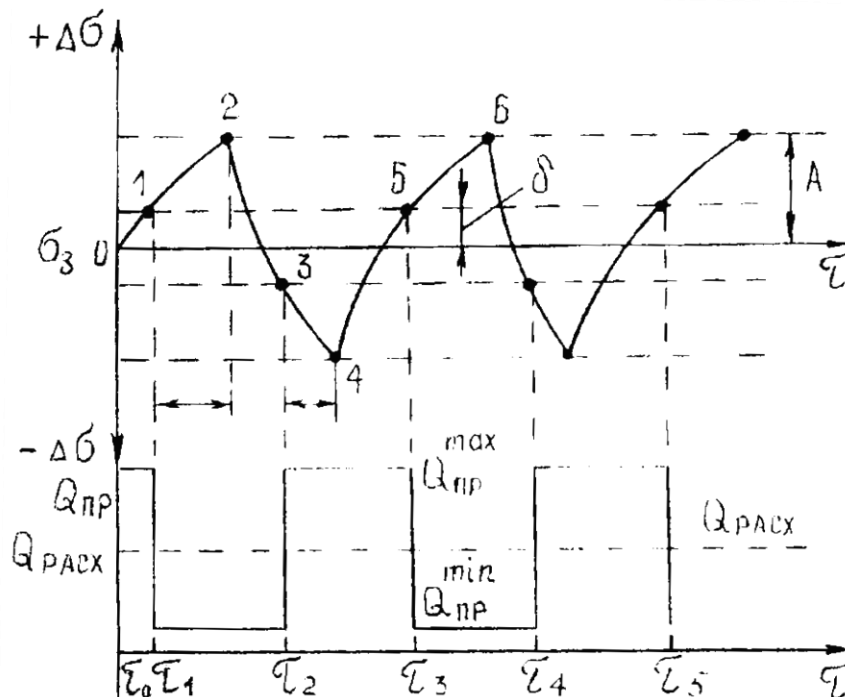


Рис. 6.17

В точці 0 $Q_{\text{под}} > Q_{\text{вит}}$ і тому параметр збільшується. В точці 1 параметр досягає верхньої точки зони нечутливості, а тому регулятор переводить регулюючий орган в положення “ЗАКР”. Але за рахунок запізнення в об’єкті параметр ще деякий час продовжує збільшуватися, а потім починає зменшуватися, тому що витрати більші за подачу. В точці 3 параметр досягає нижньої межі зони нечутливості і регулятор переводить регулюючий орган в положення “ВІДК”. За рахунок запізнення параметр ще деякий час продовжує зменшуватися, а потім починає збільшуватися, тому що подача більша за витрати.

Таким чином в системі встановлюються незатухаючі коливання параметру з амплітудою A і з частотою, яка залежить від динамічних властивостей об’єкта. Регулюючий орган з певною частотою переводиться в положення “ВІДК” та “ЗАКР”.

Максимальне відхилення параметру A визначається властивостями об’єкта і регулятора. Час зміни параметра від точки 1 до точки 2 (від точки 3 до точки 4) є величиною запізнення в об’єкті. Тому амплітуда відхилення A буде визначатися швидкістю зміни параметра, яка буде тим більшою чим менша постійна часу об’єкта. Величина A також залежить від ширини зони нечутливості, яку можна змінювати при налагодженні регулятора. Але зменшення зони нечутливості збільшує частоту переключення регулюючого органу. Між тим кожен виконавчий механізм має свою допустиму частоту переключення. Якщо виконавчим механізмом буде електричний двигун, то його часте переключення призведе до перегріву, тому що пускові струми в 5 - 8 разів перевищують номінальні. Допустима частота включення існує і для реле. В даному випадку перевагу будуть мати тиристори, допустима частота переключення яких може бути більшою за 1кГц.

Таким чином двопозиційні регулятори можуть забезпечити необхідну якість регулювання на об’єктах з малою величиною запізнення та великою постійною часу.

Колівання параметру навколо заданого значення будуть симетричні (середнє значення буде дорівнювати заданому) лише тоді, коли витримується співвідношення:

$$Q_{\text{вит}}^n = \frac{Q_{\text{под}}^{\text{мак}} - Q_{\text{под}}^{\text{мін}}}{2}.$$

В цьому випадку середнє значення параметру дорівнює заданому. Коли витрати більші за вказану величину, середнє значення параметру менше за задане. Коли витрати менші за вказану величину, середнє значення параметру більше заданого.

Позиційні регулятори характеризуються конструктивною простотою та невисокою вартістю. Але їх застосування виправдане лише тоді, коли вони можуть забезпечити необхідну якість регулювання.

6.7. Вибір регуляторів в програмі MATLAB

В цьому підрозділі вибір структури та параметрів регуляторів виконується на основі аналізу перехідної характеристики замкнутої системи. Така характеристика обчислюється для кожного варіанту побудови регулятора методом математичного моделювання за допомогою MATLAB.

Розглянемо характеристики П-, І- та Д-регуляторів і ознайомимось з методами отримання потрібного відгуку системи. Ці методи включають вибір певного типу регулятора. Система з одиничним зворотним зв'язком (система основного типу) зображена на рис. 6.18.

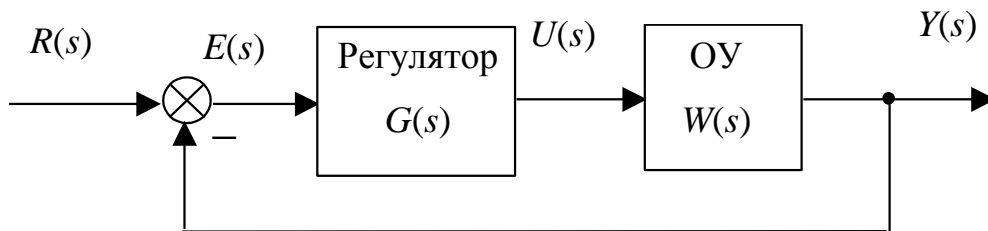


Рис. 6.18

Об'єкт управління (ОУ) представляє собою так звану А-систему, яка має керуватися певним регулятором. Регулятор забезпечує збудження об'єкта керування та керує поведінкою всієї системи. В повному обсязі регулятор складається з трьох елементів:

- пропорційного підсилювача з масштабуючим коефіцієнтом k_p ;
- інтегратора з коефіцієнтом перетворення k_i ;
- обчислювача похідної з коефіцієнтом перетворення k_d .

Передаточна функція регулятора в повному складі дорівнює

$$W(s) = k_p + \frac{k_i}{s} + k_d s = \frac{k_d s^2 + k_p s + k_i}{s}.$$

Проаналізуємо роботу ПІД-регулятора в замкнутому колі системи, зображеної на рис. 6.18. Похибка на виході суматора $e(t)$ (різниця між вхідним впливом $r(t)$ та вихідним сигналом $y(t)$) надходить до ПІД-регулятора, який її масштабує, обчислює похідну та інтеграл від неї і таким чином утворює сигнал керування $u(t)$:

$$u(t) = k_p e(t) + k_i \int e(t) dt + k_d \frac{de(t)}{dt}.$$

Сигнал керування надходить до об'єкта керування, де під впливом цього сигналу створюється нове значення вихідного сигналу $y(t)$. Це нове значення надходить до суматора, де створюється нове значення сигналу похибки $e(t)$. Регулятор отримує це нове значення сигналу похибки і виконує нові обчислення для формування нового значення сигналу керування. Така робота в замкнутому колі буде проходити до того часу, коли зникне різниця між вхідним та вихідним сигналами.

Вплив кожного із елементів регулятора на якість перехідних процесів можна відобразити за допомогою таблиці 6.1.

Таблиця 6.1

Коефіцієнт регулятора	Час досягнення максимуму	Перерегулювання	Час регулювання	Похибка сталого режиму
k_p	Зменшується	Збільшується	Невеликі зміни	Зменшується
k_i	Зменшується	Збільшується	Збільшується	Ліквідується повністю
k_d	Невеликі зміни	Зменшується	Зменшується	Невеликі зміни

Необхідно відмітити, що наведені в таблиці властивості перехідних характеристик не можуть бути зовсім точно відтворені, тому що складові елементи регулятора впливають один на одного. Фактично зміни параметрів одного елемента можуть обумовлювати зміну дії двох інших елементів. Тому таблицю слід використовувати тільки для приблизних розрахунків при синтезі регулятора.

Розглянемо просту схему об'єкту управління: деяка маса m утримується пружиною p коефіцієнтом жорсткості k та демпфуючим пристроєм з коефіцієнтом демпфування b . Рівняння математичної моделі об'єкту записується як рівняння рівноваги:

$$m \frac{d^2x}{dt^2} + b \frac{dx}{dt} + kx = f,$$

де f – зовнішня сила, що діє на об'єкт; x – поточна горизонтальна координата об'єкту.

Перетворимо цей вираз за Лапласом, що дає:

$$ms^2X(s) + bsX(s) + kX(s) = F(s).$$

На основі останньої формули запишемо передаточну функцію відносно поточної координати $X(s)$ (вихід об'єкта) та сили $F(s)$ (вхід об'єкта):

$$\frac{X(s)}{F(s)} = \frac{1}{ms^2 + bs + k}.$$

Будемо вважати, що $m = 1$ кг; $b = 10$ Н·с/м; $k = 20$ Н/м; $f = 1$ Н. Тоді отримуємо

$$\frac{X(s)}{F(s)} = \frac{1}{s^2 + 10s + 20}.$$

При вирішенні задачі синтезу САК необхідно показати, що кожний із типів регуляторів (пропорційний, інтегральний та диференційний) сприяє:

- підвищенню швидкодії системи;
- мінімізації перерегулювання ;
- зменшенню похибки сталого режиму.

6.8. Синтез регуляторів методом корневих годографів

Розглянемо структурну схему системи з одиничним зворотнім зв'язком (систему основного типу) та з відокремленим в ній пропорційним регулятором (рис. 6.19).

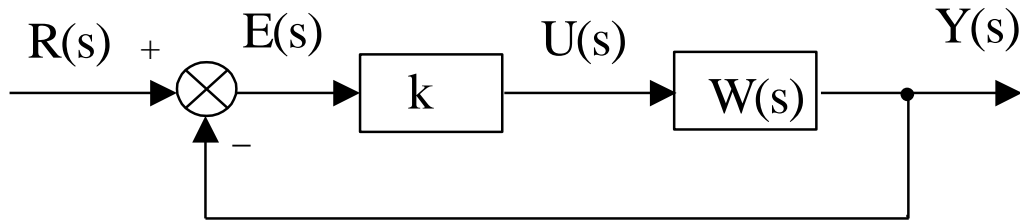


Рис. 6.19

Будемо вважати, що коефіцієнт підсилення регулятора дорівнює k , а передаточна функція має загальний вигляд, тобто:

$$W(s) = \frac{\sum_{i=0}^m b_i s^i}{\sum_{j=0}^n a_j s^j}.$$

Тоді передаточна функція замкнутого кола системи буде дорівнювати

$$\frac{Y(p)}{R(p)} = \frac{kW(p)}{1 + kW(p)}.$$

Як відомо [24 – 26], корені рівняння, що відповідає поліному чисельника передаточної функції $W(s)$, мають назву “нулі”, а корені рівняння, яке відповідає знаменнику передаточної функції $W(s)$, мають назву “полюси”. Ці поняття переносяться також на передаточні функції замкнутих зворотнім зв'язком систем, тобто нулі є рішеннями рівняння

$$kW(s)=0,$$

а полюси – це рішення рівняння

$$1+kW(s)=0.$$

Корневим годографом системи з $W(s)$ є геометричне місце точок (графік розміщення) усіх можливих полюсів замкнутого кола з пропорційним коефіцієнтом підсилення k і від'ємним одиничним зворотнім зв'язком, k змінюється від 0 до ∞ .

Позначимо поліноми чисельника та знаменника передаточної функції розімкнутої системи так, що

$$W(s) = \frac{P(s)}{Q(s)}.$$

Передаточна функція замкнутої системи:

$$\frac{Y(s)}{R(s)} = \frac{kP(s)}{Q(s) + kP(s)}.$$

Тоді можна записати характеристичне рівняння замкнутої системи:

$$Q(s) + kP(s) = 0,$$

або

$$\frac{Q(s)}{k} + P(s) = 0.$$

В даному випадку має виконуватися умова $n \geq m$, яка називається умовою фізичної реалізуємості системи (система зі зворотнім зв'язком еквівалентна фільтру низьких частот).

Розглянемо можливі значення коефіцієнта підсилення k . Якщо $k \rightarrow 0$, то полюси замкнутої системи співпадають з полюсами розімкнутої системи, тобто характеристичне рівняння замкнутої системи $Q(s) = 0$. Якщо $k \rightarrow \infty$, то полюси замкнутої системи співпадають з нулями передаточної функції розімкнутої системи, тобто характеристичне рівняння замкнутої системи $P(s) = 0$. Незалежно від величини k замкнута система завжди має n полюсів, причому n – це число полюсів $W(s)$. Таким чином, кореневий годограф повинен мати n віток і кожна з віток починається від полюса $W(s)$ та рухається до нуля $W(s)$. Якщо $W(s)$ має більше полюсів чім нулів, що має місце в більшості випадків, то $m < n$ і це означає, що $W(s)$ має нулі на нескінченості. В цьому випадку

$$\lim_{s \rightarrow \infty} W(s) = 0.$$

Число нулів при $s \rightarrow \infty$ дорівнює $(n-m)$ (“число полюсів мінус число нулів”) і є числом гілок кореневого годографу, які прямують до нескінченості (є асимптотами). Таким чином, кореневий годограф – це фактично розташування усіх можливих полюсів замкнутого кола. Тому на основі кореневого годографу можна вибирати такий коефіцієнт підсилення k , який забезпечує технічні вимоги до параметрів перехідного процесу замкнутої системи. Якщо будь-який з відібраних полюсів знаходиться в правій напівплощині, то замкнута система буде нестійкою. Полюси, що розташовані найближче до мнимої вісі, мають найбільший вплив на параметри перехідного процесу (вони забезпечують найбільші постійні часу $T_i = -1/\text{Re}(s_i)$). Якщо замкнута система має 3, 4 або більше полюсів, то в цьому випадку вона буде поводитися аналогічно системі 2-го або навіть 1-го порядку в залежності від конкретного розташування вказаних полюсів, які мають назву домінуючих.

6.9. Синтез регуляторів частотними методами

Частотний метод синтезу регуляторів дещо менше відповідає закономірностям протікання фізичних процесів в системі, що проектується. Але він має суттєві переваги перед іншими методами, особливо при

моделюванні передаточних функцій на основі фізичних даних. Частотний метод має дві модифікації: на основі ЛАХ (діаграма Боде по термінології MATLAB); на основі АФХ на комплексній площині (діаграма Найквіста за термінологією MATLAB). Як відомо [25, 26], ці два методи відображають одну і ту ж саму інформацію, а різниця між методами полягає в формі представлення цієї інформації.

Відомо, що вихідний сигнал лінійної системи при синусоїдальному вхідному впливі є синусоїда тієї ж частоти з фазовим зсувом, що вносить система. Тому частотний відгук визначається як амплітудні та фазові різниці поміж вхідними та вихідними синусоїдами. При цьому можна використовувати частотний відгук розімкнутої системи для передбачення її поведінки в замкнутому стані (при замиканні зворотного зв'язку).

Для побудови частотних характеристик створюється вектор частот, який має охопити діапазон від нуля до нескінченості, та обчислюється величина передаточної функції на цих частотах. Якщо $W(s)$ – передаточна функція розімкнутого кола системи, а ω – вектор частоти, то будується графік $W(j\omega)$ відносно ω . В зв'язку з тим, що $W(j\omega)$ – функція комплексного числа, то для неї можна побудувати графіки модуля та фази. В результаті отримуємо діаграму Боде. Якщо розташувати цю функцію на комплексній площині, то отримуємо діаграму Найквіста.

6.10. Синтез регуляторів на основі моделювання об'єктів та систем в просторі стану

Як відомо [42 – 44], в просторі стану математичні моделі систем задаються двома рівняннями:

$$\frac{dX(t)}{dt} = AX(t) + BU(t),$$

$$Y(t) = CX(t) + DU(t),$$

де X – вектор стану системи розмірності $(n \times 1)$, в електромеханічних системах це змінні положення та швидкості;

U – вектор розмірності $(r \times 1)$ (може бути скаляр), що визначає вхідний вплив (сила як причина деякого руху, момент сили, обертаючий момент в електромеханічних системах);

Y – вектор розмірності $(m \times 1)$ (може бути скаляр) вихідного сигналу;

A – матриця стану (матриця динаміки системи) розмірністю $(n \times n)$;

B – матриця керування розмірністю $(n \times r)$;

C – матриця вимірів розмірністю $(m \times n)$;

D – матриця розмірністю $(m \times r)$, що зв'язує вхід та вихід системи;

n – кількість змінних стану системи;

r – кількість входів системи;

m – кількість виходів системи.

Ці вектори та матриці визначають змінні стану, змінні входу та виходу об'єкту керування або системи. Перше рівняння приховує в собі п диференційних рівнянь першого порядку, перше із яких визначає певну координату, друге – швидкість зміни цієї координати, третє – третю похідну від координати і так далі. В просторі стану можуть моделюватися системи з багатьма входами та виходами (так звані MIMO – системи), або з одним входом та одним виходом (так звані SISO – системи). В прикладах проектування систем управління як правило використовується в основному системи з одним входом та одним виходом.

6.11. Методи оцінювання параметрів руху об'єктів вимірювань

У задачі автоматизованої обробки інформації про рух об'єктів вимірювань (ОВ) найбільш часто використовуються рекурентні алгоритми оцінювання параметрів руху. Ці алгоритми забезпечують високу швидкість і економічні за обсягом виконуваних обчислень і пам'яті комп'ютера, що використовується. Крім того, такі алгоритми найкращим чином відповідають вимогам, які висуваються до алгоритмів контролю за параметрами руху об'єктів. Вони ґрунтуються на процедурах прогнозу і подальшої корекції значень поточних координат об'єкту за результатами траєкторних вимірювань.

Апробованим рекурентним алгоритмом оцінки параметрів руху є фільтр Калмана [36 - 38]. Вектор стану ОВ (виробничого обладнання та промислових виробів, що виготовляються) для отримання оцінок параметрів руху за допомогою фільтра Калмана запишемо у вигляді:

$$Z(t) = [r(t) v(t) a(t)]^T,$$

де $r(t), v(t), a(t)$ – значення поточних координат, швидкості та прискорення ОВ.

Дискретною математичною моделлю руху елементів виробничого обладнання в разі відсутності перешкод є різницеве рівняння:

$$Z_j = \Phi_{j-1} Z_{j-1},$$

де Z_j - значення вектора стану в дискретні моменти часу, що відповідають сформованим кадрам часової відеопослідовності, Φ_{j-1} - передаточна матриця (матриця динаміки ОВ).

Наприклад, для поступального або обертового руху з постійним прискоренням та кроком дискретності спостережень $\delta_\delta = const$ маємо:

$$\Phi_{j-1} = \Phi = \begin{bmatrix} 1 & \delta_\delta & \delta_\delta^2/2 \\ 0 & 1 & \delta_\delta \\ 0 & 0 & 1 \end{bmatrix}.$$

Крок дискретності спостережень обумовлено, наприклад, використанням для вимірювань ряду відеозображень (кадрів), що складають часову

послідовність відеозображень з вимірювальною інформацією про геометричні параметри та параметри руху ОБ.

У реальних процесах руху елементів виробничого обладнання завжди мають місце випадкові перешкоди, обумовлені зовнішніми впливами:

$$\Lambda(t) = [\lambda_r(t) \lambda_v(t) \lambda_a(t)]^T,$$

де $\lambda_r(t), \lambda_v(t), \lambda_a(t)$ - випадкові перешкоди по поточній координаті, швидкості і прискорення відповідно.

Тому для реальних процесів руху елементів виробничого обладнання

$$Z_j = \Phi_{j-1} Z_{j-1} + \Pi_{j-1} \Lambda_{j-1},$$

де Π_{j-1} - матриця, що визначає вплив складових частин вектора випадкових завад $\Lambda(t)$ на елементи вектора стану Z_j .

Для поступального або обертового руху з постійним прискоренням при $\delta_\delta = const$:

$$\Pi_{j-1} = \Pi = \begin{bmatrix} 1 & \delta_\delta & \delta_\delta^2/2 \\ 0 & 1 & \delta_\delta \\ 0 & 0 & 1 \end{bmatrix}.$$

На основі дискретного фільтра Калмана можуть бути отримані оцінки параметрів руху елементів виробничого обладнання у вигляді:

$$Z_{j/j} = Z_{j/(j-1)} + \Gamma_j (\alpha_j^* - H Z_{j/(j-1)});$$

$$Z_{j/(j-1)} = \Phi_{(j-1)} Z_{(j-1)/(j-1)};$$

$$\Gamma_j = P_{j/(j-1)} H^T (H P_{j/(j-1)} H^T + \Psi_j)^{-1};$$

$$P_{j/(j-1)} = \Phi_{(j-1)} P_{(j-1)/(j-1)} \Phi_{(j-1)}^T + \Pi_{j-1} Q_j \Pi_{j-1}^T;$$

$$P_{j/j} = P_{j/(j-1)} (I - \Gamma_j H),$$

де $Z_{j/(j-1)}$ - прогнозована оцінка вектора стану для моменту часу $j\delta_\delta$ на основі результатів вимірювань координат від початку вимірювання до моменту часу $(j-1)\delta_\delta$; $Z_{j/j}$ - уточнена оцінка вектора стану для моменту часу $j\delta_\delta$ на основі результатів вимірювань координат від початку вимірювань до моменту часу $(j-1)\delta_\delta$; Γ_j - матриця коефіцієнтів посилення фільтра Калмана; H_j - матриця вимірювань; $P_{j/(j-1)}$ - коваріаційна матриця похибок прогнозованої оцінки вектора стану; $P_{j/j}$ - коваріаційна матриця уточненої оцінки вектора стану; Ψ_j - коваріаційна матриця похибок вимірювань для дискретних відліків вихідних величин об'єкта вимірювань; Q_j - коваріаційна матриця дискретних відліків випадкових завад Λ_j для реального процесу руху об'єкта вимірювання; I - одинична матриця.

Сама оцінка вектора станів на основі фільтра Калмана виконується за рекурентною процедурою в два етапи:

- прогноз вектора стану;
- корекція (оцінка) вектора стану.

Розглянуті методи оцінки параметрів руху на основі фільтра Калмана використовують інформацію про рух елементів виробничого обладнання, накопичену за весь інтервал спостереження. Так як виробниче обладнання часто змінює свої параметри руху, то оцінки на основі фільтра Калмана містять значні динамічні похибки.

У зв'язку із зазначеними недоліками виникає необхідність побудови рекурентного алгоритму, який використовує фіксований обсяг вибірки попередніх значень параметрів руху [27, 28, 35]. Такий алгоритм забезпечує послідовне (на кожному кроці) уточнення параметрів руху шляхом послідовного згладжування результатів вимірювань.

Для випадку оцінювання однієї координати суть методу послідовного згладжування полягає в наступному. За результатами фіксованої кількості попередніх вимірювань координати отримуємо оцінки параметрів руху (координат, швидкостей, прискорень) $\hat{r}_{j-1}, \hat{v}_{j-1}, \hat{a}_{j-1}$ на момент часу t_{j-1} . Далі екстраполюємо ці значення на момент t_j наступного вимірювання. При цьому в момент t_j виробляється нове вимірне значення координати. Завдання полягає в тому, щоб за відомими екстрапольовані і вимірним значенням координати r отримати уточнене згладжені значення \hat{r}_j цієї координати на момент часу t_j .

Для параметра руху x (це може бути поточна координата, швидкість або прискорення об'єкта вимірювань) розрахунок експоненціального згладженого значення здійснюється за формулою [27, 35]:

$$\hat{x}_j = (1 - \xi)x_j + \xi f[\hat{x}_{j-1}, \hat{x}_{j-2}, \dots, \hat{x}_{j-s}], \quad (6.1)$$

де x_j - вимірне значення параметра в j -м кадрі відеопослідовності;
 \hat{x} - згладжене значення параметра в j -м кадрі відеопослідовності;
 $f[]$ - деяка функція попередніх згладжених значень параметра руху;
 ξ - постійна величина, що має фізичний зміст коефіцієнта згладжування,
 s - кількість попередніх значень параметра руху, що використовується для екстраполяції.

У найпростішому випадку для знаходження наступного загладжуваного значення параметру руху використовується тільки поточне вимірне значення та попереднє згладжене значення. Тоді, формула (6.1) приймає наступний вигляд [27]:

$$\hat{\theta}_n = (1 - \xi)\theta_n + \xi\hat{\theta}_{n-1}. \quad (6.2)$$

Формулу (6.2) можна узагальнити та записати у вигляді:

$$\hat{\theta}_n = (1-\xi)\theta_n + \xi(1-\xi)\theta_{n-1} + \xi^2(1-\xi)\theta_{n-2} + \dots + \xi^n(1-\xi)\theta_0 = \sum_{i=0}^n \eta_i \theta_{n-i}, \quad (6.3)$$

де $\eta_i = \xi^i(1-\xi)$ - вагова функція, що враховує поточне та попередні виміряні значення параметрів руху ОУ.

Чим менше значення коефіцієнта ξ в формулах (6.2) і (6.3), тим в меншій степені враховуються результати попередніх вимірювань (менше згладжування), та навпаки, чим більше значення коефіцієнта ξ , тим більше згладжування, тобто результати попередніх вимірювань параметрів руху враховуються в більшій мірі при обчисленні згладжених поточних значень цих параметрів. Відповідно, коефіцієнт ξ називається експоненціальним коефіцієнтом згладжування.

Для параметрів руху, що змінюються за лінійним законом, оцінка на основі експоненціального згладжування набуває вигляду [27, 35]:

$$\hat{\theta}_n = (1-\xi)\theta_n + \xi f[\hat{\theta}_{n-1}, \hat{\theta}_{n-2}].$$

Обчислювальна операція $f[...]$ над попередніми згладженими значеннями параметрів руху забезпечує знаходження екстрапольованих значень параметрів руху на момент t_n останнього вимірювання цих параметрів.

Ефективність застосування експоненціального згладжування до оцінки траєкторій та параметрів руху об'єктів оцінюється за величиною випадкових та динамічних похибок згладжування. Сумарну похибку згладжування параметра можна представити в наступному вигляді [27]:

$$\Delta\theta_{n\Sigma} = \sum_{i=0}^n \eta_i (\theta_{n-i}^{(0)} + \Delta\theta_{n-i}) - \theta_n^{(0)} = \left(\sum_{i=0}^n \eta_i \theta_{n-i}^{(0)} + \Delta\theta_{n-i} \right) + \sum_{i=0}^n \eta_i \theta_{n-i}^{(0)}. \quad (6.4)$$

Перший доданок (виділений дужками) у формулі (6.4) є динамічною складовою $\Delta_\delta \hat{\theta}_n$ сумарної похибки згладжування, а другий доданок – є випадковою складовою $\Delta_\delta \hat{\theta}_{ec}$ цієї похибки.

Так як випадкова і динамічна складові похибки не корельовані між собою, дисперсія сумарної похибки дорівнює:

$$\sigma_\Sigma^2 = \sigma_{\hat{\theta}_n}^2 + (\Delta_\delta \hat{\theta}_n)^2. \quad (6.5)$$

В [27] обчислено дисперсію випадкової складової похибки визначення параметрів руху ОВ та отримано формулу:

$$\sigma_{\hat{\theta}_n}^2 = \sigma_\theta^2 \frac{(1-\xi)^2}{1+\xi}. \quad (6.6)$$

Формула (6.6) вказує на той факт, що дисперсія випадкової похибки експоненціального згладжування параметрів руху зменшується зі збільшенням коефіцієнту згладжування.

В [27] також обчислено квадрат динамічної складової похибки визначення параметрів руху ОВ:

$$(\Delta_{\sigma} \hat{\theta}_n)^2 = \frac{\xi^2}{(1-\xi)^2} (\Delta^{(1)} \theta_n)^2. \quad (6.7)$$

Формула (6.7) вказує на той факт, що квадрат динамічної складової похибки експоненціального згладжування параметрів руху збільшується зі збільшенням коефіцієнту згладжування.

З урахуванням формул (6.5) – (6.7) дисперсія відносної сумарної помилки згладжувань дорівнює [27]

$$\frac{\sigma_{\Sigma}^2}{\sigma_{\theta}^2} = \frac{(1+\xi)^2}{(1-\xi)} + \frac{\xi^2}{(1-\xi)^2} s^2, \quad (6.8)$$

де $s^2 = \frac{(\Delta^{(1)} \theta_n)^2}{\sigma_{\theta}^2}$ – відносна інтенсивність можливих змін параметрів руху

ОВ певного класу.

Для знаходження оптимального значення ξ_{opt} за критерієм мінімуму дисперсії сумарної помилки можна продиференціювати формулу (6.8) по ξ і прирівняти результат до нуля. В [27] рівняння для знаходження оптимального значення коефіцієнту згладжування ξ_{opt} має вигляд:

$$(1-\xi_{opt}^2)(1-\xi_{opt})^3 + 2(1-\xi_{opt})^4 - 2\xi_{opt}(1+\xi_{opt})^2 s^2 = 0. \quad (6.9)$$

Використовуючи рівняння (6.9), можна вибрати оптимальне значення коефіцієнту згладжування ξ_{opt} , якщо відома інтенсивність змін параметрів руху ОВ певного класу.

Експоненціальне згладжування дає більш точну оцінку поточних координат та параметрів руху об'єктів у порівнянні з методами чисельного диференціювання. Також експоненціальне згладжування дає більш точний результат в порівнянні з фільтром Калмана, якщо елементи виробничого обладнання змінюють свої параметри руху на інтервалі спостереження.

6.12. Визначення координат і параметрів руху об'єктів в реальному часі

Використання засобів вимірювання механічних величин на промислових підприємствах потребує оцінки поточних координат, геометричних параметрів та параметрів руху ОВ у реальному часі. На основі часових послідовностей відеозображень ОВ безпосередньо можуть бути визначені тільки поточні значення координат та кутового положення цих ОВ. Параметри руху (лінійні і кутові швидкості і прискорення) повинні бути ідентифіковані на основі алгоритмічної обробки результатів вимірювань координат і кутового положення. Для цього потрібно розробити алгоритмічні методи ідентифікації, що мають високу швидкодію і здатні формувати оцінки параметрів руху в реальному часі.

Одним з можливих варіантів отримання оцінок параметрів руху у реальному часі є фільтр Калмана [32, 33, 42, 36 - 38]. Такий підхід дозволяє уникнути операцій чисельного диференціювання результатів вимірювань

поточних значень координат і суттєво підвищити точність і швидкодію процедур визначення параметрів руху.

Вектор стану ОВ запишемо у вигляді:

$$Z(t) = [x(t) \quad v(t) \quad a(t)]^T. \quad (6.10)$$

Дискретною математичною моделлю руху ОВ у випадку відсутності збурень є різницеве рівняння [32, 33]

$$Z_i = \Phi_{i-1} Z_{i-1}, \quad (6.11)$$

де Φ_{i-1} – перехідна матриця (матриця динаміки ОВ).

Наприклад, для поступального або обертального руху з постійним прискоренням при $\delta_d = const$ маємо:

$$\Phi_{i-1} = \Phi = \begin{bmatrix} 1 & \delta_d & \delta_d^2/2 \\ 0 & 1 & \delta_d \\ 0 & 0 & 1 \end{bmatrix}. \quad (6.12)$$

В реальних процесах руху ОВ завжди мають місце випадкові збурення $\Lambda(t) = [\lambda_x(t) \quad \lambda_v(t) \quad \lambda_a(t)]^T$, які обумовлені:

– дією зовнішніх збурюючих впливів на ОВ, що приводить до похибок координати $\lambda_x(t)$ відносно ідеальної математичної моделі (6.10), (6.11);

– нестабільністю підтримання параметрів руху за допомогою системи керування ОВ, наприклад наявністю похибки підтримання швидкості руху або частоти обертання $\lambda_v(t)$;

– обмеженням порядку похідної ($a(t) = a_0 = const$) в математичній моделі руху (6.11) з перехідною матрицею (6.12), тому для математичної моделі реальних процесів руху $a(t) = a_0 + \lambda_a(t) \neq const$.

Тому для реальних процесів руху ОВ [32, 33]

$$Z_i = \Phi_{i-1} Z_{i-1} + \Pi_{i-1} \Lambda_{i-1}, \quad (6.13)$$

де Π_{i-1} – матриця, що визначає вплив складових частин вектора випадкових збурень $\Lambda(t)$ на елементи вектора стану Z_i . Для поступального або обертального руху з постійним прискоренням при $\delta_d = const$

$$\Pi_{i-1} = \Pi = \begin{bmatrix} 1 & \delta_d & \delta_d^2/2 \\ 0 & 1 & \delta_d \\ 0 & 0 & 1 \end{bmatrix}, \quad \Lambda_{i-1} = \Lambda(\delta_d \cdot (i-1)). \quad (6.14)$$

Для кожного відеозображення вимірюється поточне значення координати ОВ

$$x_i^* = H_i Z_i + \Delta_{xi}, \quad (6.15)$$

де H_i – матриця вимірювань, в даному випадку $H_i = H = [1 \quad 0 \quad 0]$, Δ_{xi} – похибка вимірювань координати ОВ для i -го відеозображення.

На основі дискретного фільтра Калмана [27, 32, 40, 41] можуть бути отримані оцінки параметрів руху ОБ у вигляді:

$$\begin{aligned}
 \hat{Z}_{i/i} &= \hat{Z}_{i/(i-1)} + \Gamma_i (x_i^* - \mathbf{H}\hat{Z}_{i/(i-1)}), \\
 \hat{Z}_{i/(i-1)} &= \Phi_{i-1} \hat{Z}_{(i-1)/(i-1)}, \\
 \Gamma_i &= P_{i/(i-1)} \mathbf{H}^T (\mathbf{H} P_{i/(i-1)} \mathbf{H}^T + \Psi_i)^{-1}, \\
 P_{i/(i-1)} &= \Phi_{i-1} P_{(i-1)/(i-1)} \Phi_{i-1}^T + \Pi_{i-1} Q_i \Pi_{i-1}^T, \\
 P_{i/i} &= P_{i/(i-1)} (I - \Gamma_i \mathbf{H}),
 \end{aligned} \tag{6.16}$$

де $\hat{Z}_{i/(i-1)}$ – прогнозована оцінка вектора стану для моменту часу $i\delta_d$ на основі результатів вимірювань координат від початку вимірювань до моменту часу $(i-1)\delta_d$, $\hat{Z}_{i/i}$ – уточнена оцінка вектора стану для моменту часу $i\delta_d$ на основі результатів вимірювань координат від початку вимірювань до моменту часу $i\delta_d$, Γ_i – матриця коефіцієнтів підсилення фільтра Калмана, $P_{i/(i-1)}$ – коваріаційна матриця похибок прогнозованої оцінки вектора стану, $P_{i/i}$ – коваріаційна матриця похибок уточненої оцінки вектора стану, Ψ_i – коваріаційна матриця похибок вимірювань для дискретних відліків вихідних величин ОБ, Q_i – коваріаційна матриця дискретних відліків випадкових збурень Λ_i для реального процесу руху ОБ, I – одинична матриця.

Матриці Ψ_i і Q_i визначаються, виходячи з властивостей ОБ та збурень, що діють на ці об'єкти. В даному випадку для рівнянь (6.15), (6.16) $\Psi_i = \sigma_{\Delta x}^2$, тобто коваріаційна матриця результатів вимірювань дорівнює дисперсії похибки дискретних відліків поточної координати ОБ. Так як для поступального або обертального руху з постійним прискоренням складові частини випадкових збурень Λ_i викликані множиною різних незалежних факторів, то вважаємо ці збурення некорельованими з нульовим середнім значенням. Тому коваріаційна матриця випадкових збурень

$$Q_i = \begin{bmatrix} \sigma_{\lambda_x}^2 & 0 & 0 \\ 0 & \sigma_{\lambda_v}^2 & 0 \\ 0 & 0 & \sigma_{\lambda_a}^2 \end{bmatrix}, \tag{6.17}$$

де $\sigma_{\lambda_x}^2$, $\sigma_{\lambda_v}^2$ і $\sigma_{\lambda_a}^2$ – дисперсії послідовностей дискретних відліків випадкових процесів $\lambda_x(t)$, $\lambda_v(t)$ і $\lambda_a(t)$ відповідно.

Для реалізації обчислень поточних оцінок вектора стану за формулами (6.16) необхідно задати початкові умови. Згідно [34] в якості початкових умов необхідно використовувати апріорні дані про ОБ:

$$\hat{Z}_0 = E[Z_0], \quad P_0 = E[Z_0 \cdot Z_0^T], \tag{6.18}$$

де Z_0 – апріорна оцінка початкового стану ОВ, $E[.]$ – математичне сподівання виразу в дужках.

Якщо досліджується рух ОВ з вектором стану (6.10), то на основі формул (6.18) отримуємо:

$$\begin{aligned} Z_0 &= [x_0 \quad v_0 \quad a_0]^T, \\ \hat{Z}_0 &= E[Z_0] = [E(x_0) \quad E(v_0) \quad E(a_0)]^T, \\ P_0 &= E[Z_0 \cdot Z_0^T] = \begin{bmatrix} E(x_0^2) & E(x_0 v_0) & E(x_0 a_0) \\ E(x_0 v_0) & E(v_0^2) & E(v_0 a_0) \\ E(x_0 a_0) & E(v_0 a_0) & E(a_0^2) \end{bmatrix} = \\ &= \begin{bmatrix} E^2(x_0) + \sigma_{x0}^2 & E(x_0)E(v_0) + \rho_{xv}\sigma_{x0}\sigma_{v0} & E(x_0)E(a_0) + \rho_{xa}\sigma_{x0}\sigma_{a0} \\ E(x_0)E(v_0) + \rho_{xv}\sigma_{x0}\sigma_{v0} & E^2(v_0) + \sigma_{v0}^2 & E(v_0)E(a_0) + \rho_{va}\sigma_{v0}\sigma_{a0} \\ E(x_0)E(a_0) + \rho_{xa}\sigma_{x0}\sigma_{a0} & E(v_0)E(a_0) + \rho_{va}\sigma_{v0}\sigma_{a0} & E^2(a_0) + \sigma_{a0}^2 \end{bmatrix}, \end{aligned}$$

де $\sigma_{x0}^2, \sigma_{v0}^2, \sigma_{a0}^2$ – дисперсії складових частин початкового стану ОВ, $\rho_{xv}, \rho_{xa}, \rho_{va}$ – коефіцієнти кореляції між цими складовими частинами.

Наприклад, якщо початкове положення ОВ в площині відеозображення вважається початком координат ($E(x_0) = 0$), початкові значення швидкості і прискорення мають симетричний розподіл відносно $E(v_0) = 0$ і $E(a_0) = 0$, початкові значення параметрів руху є некорельованими між собою ($\rho_{xv} = \rho_{xa} = \rho_{va} = 0$), то

$$\begin{aligned} \hat{Z}_0 &= E[Z_0] = [0 \quad 0 \quad 0]^T, \\ P_0 &= E[Z_0 \cdot Z_0^T] = \begin{bmatrix} \sigma_{x0}^2 & 0 & 0 \\ 0 & \sigma_{v0}^2 & 0 \\ 0 & 0 & \sigma_{a0}^2 \end{bmatrix}. \end{aligned} \quad (6.19)$$

Розглянемо питання точності оцінок параметрів руху на основі фільтра Калмана. Діагональні елементи коваріаційної матриці

$$P_{i/i} = E[(\hat{Z}_{i/i} - Z_i)(\hat{Z}_{i/i} - Z_i)^T]$$

визначають поточну похибку оцінки відповідних елементів вектора стану.

Для дискретного фільтра Калмана коваріаційна матриця $P_{i/i}$ розраховується на основі рекурентних співвідношень (6.16). Тому аналітичний вираз похибок оцінки вектора стану є досить складним.

Іншим варіантом визначення точності фільтра Калмана є отримання рівнянь неперервного фільтра з рівнянь дискретного фільтра при $\delta_d \rightarrow 0$ [] та визначення для нього діагональних елементів коваріаційної матриці $P(t)$ в статичному режимі [33, 43, 44]. Розглянемо цей варіант для випадку рівноприскореного руху ОВ.

Неперервна математична модель рівноприскореного руху ОВ може бути отримана з дискретної моделі [42] і задається рівняннями

$$\begin{aligned} \dot{Z}(t) &= \Phi_H(t)Z(t) + \Pi_H(t)\Lambda(t), \\ x^*(t) &= H_I Z(t) + \Delta_x(t), \end{aligned} \quad (6.20)$$

де

$$\Phi_H(t) = \Phi_H = \lim_{\delta_d \rightarrow 0} ((\Phi - I) / \delta_d) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} - \quad (6.21)$$

– матриця динаміки ОВ,

$$\Pi_H(t) = \Pi_H = \lim_{\delta_d \rightarrow 0} ((\Pi - I) / \delta_d) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} - \quad (6.22)$$

– матриця, що визначає вплив складових частин вектора випадкових збурень $\Lambda(t)$ на елементи вектора стану $Z(t)$,

$$H_I = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} - \quad (6.23)$$

– матриця вимірювань, $\Delta_x(t)$ – похибка вимірювань.

Коваріаційна матриця похибок оцінювання $P(t)$ для математичної моделі (6.20) визначається на основі рішення матричного диференційного рівняння, що є рівнянням Ріккати [77]:

$$\dot{P}(t) = \Phi_H(t)P(t) + P(t)\Phi_H^T(t) - P(t)H_H^T\Psi_H^{-1}(t)H_H P(t) + \Pi_H(t)Q_H(t)\Pi_H^T(t), \quad (6.24)$$

де $\Psi_i(t)$ – коваріаційна матриця похибок вимірювань вихідних величин ОВ,

$Q_i(t)$ – коваріаційна матриця випадкових збурень $\Lambda(t)$ для неперервної математичної моделі руху ОВ.

Будемо вважати, що шум вимірювань координати та випадкові збурення вектора стану в неперервному та дискретному випадках мають однакову спектральну щільність потужності [42]. Тоді дисперсія похибки оцінки вектора стану при $T_c \rightarrow \infty$, $\delta_d \rightarrow 0$

$$\lim_{\delta_d \rightarrow 0} D[\Delta \hat{Z}_{i/i}] = \lim_{\delta_d \rightarrow 0} P_{i/i} = \lim_{\delta_d \rightarrow 0} P_{i/(i-1)} = P(t),$$

а коваріаційні матриці

$$\Psi_H(t) = \Psi_i \cdot \delta_d = \sigma_{\Delta x}^2 \cdot \delta_d, \quad Q_H(t) = Q_i \cdot \delta_d = \begin{bmatrix} \sigma_{\lambda x}^2 \cdot \delta_d & 0 & 0 \\ 0 & \sigma_{\lambda y}^2 \cdot \delta_d & 0 \\ 0 & 0 & \sigma_{\lambda a}^2 \cdot \delta_d \end{bmatrix}. \quad (6.25)$$

Підставляючи формули (6.21) – (6.23), (6.25) в рівняння (6.24) та враховуючи те, що коваріаційна матриця похибок оцінювання є симетричною, а її похідна в статичному режимі дорівнює нулю, отримуємо систему рівнянь:

$$\left\{ \begin{array}{l} 2P_{12} - \frac{P_{11}^2}{\sigma_{\Delta x}^2 \cdot \delta_d} + \sigma_{\lambda v}^2 \cdot \delta_d = 0, \\ P_{22} + P_{13} - \frac{P_{11}P_{12}}{\sigma_{\Delta x}^2 \cdot \delta_d} = 0, \\ P_{23} - \frac{P_{11}P_{13}}{\sigma_{\Delta x}^2 \cdot \delta_d} = 0, \\ 2P_{23} - \frac{P_{12}^2}{\sigma_{\Delta x}^2 \cdot \delta_d} + \sigma_{\lambda a}^2 \cdot \delta_d = 0, \\ P_{33} - \frac{P_{12}P_{13}}{\sigma_{\Delta x}^2 \cdot \delta_d} = 0, \\ \frac{P_{13}^2}{\sigma_{\Delta x}^2 \cdot \delta_d} = 0, \end{array} \right. \quad (6.26)$$

де P_{lk} – елементи кореляційної матриці похибок оцінювання в статичному режимі.

Вирішимо систему рівнянь (6.26). З шостого, третього та п'ятого рівнянь отримуємо відповідно $P_{13} = 0$, $P_{23} = 0$, $P_{33} = 0$. З урахуванням цього перше, друге та четверте рівняння набувають вигляду:

$$\left\{ \begin{array}{l} 2P_{12} - \frac{P_{11}^2}{\sigma_{\Delta x}^2 \cdot \delta_d} + \sigma_{\lambda v}^2 \cdot \delta_d = 0, \\ P_{22} - \frac{P_{11}P_{12}}{\sigma_{\Delta x}^2 \cdot \delta_d} = 0, \\ -\frac{P_{12}^2}{\sigma_{\Delta x}^2 \cdot \delta_d} + \sigma_{\lambda a}^2 \cdot \delta_d = 0. \end{array} \right. \quad (6.27)$$

З третього рівняння системи (6.27) отримуємо $P_{12} = \sigma_{\Delta x} \sigma_{\lambda a} \cdot \delta_d$. З першого рівняння отримуємо $P_{11} = \sigma_{\Delta x} \cdot \delta_d \sqrt{2\sigma_{\lambda a} \sigma_{\Delta x} + \sigma_{\lambda v}^2}$. З другого рівняння отримуємо $P_{22} = \sigma_{\lambda a} \cdot \delta_d \sqrt{2\sigma_{\lambda a} \sigma_{\Delta x} + \sigma_{\lambda v}^2}$.

Таким чином,

$$\begin{aligned} D_{\text{ФК}}[\Delta \hat{x}_j] &= P_{11} = \sigma_{\Delta x} \cdot \delta_d \sqrt{2\sigma_{\lambda a} \sigma_{\Delta x} + \sigma_{\lambda v}^2}, \\ D_{\text{ФК}}[\Delta \hat{v}_j] &= P_{22} = \sigma_{\lambda a} \cdot \delta_d \sqrt{2\sigma_{\lambda a} \sigma_{\Delta x} + \sigma_{\lambda v}^2}, \\ D_{\text{ФК}}[\Delta \hat{a}_j] &= P_{33} = 0. \end{aligned} \quad (6.28)$$

Порівняємо отримані результати з точністю оцінок за методом чисельного диференціювання при $\Lambda(t) \neq 0$. Випадкові збурення визначаються за формулами (6.13), (6.14) і (6.17). Для цього методу дисперсія оцінок параметрів руху при $q=1$ дорівнює:

$$D_{\text{чД}}[\Delta\hat{x}_j] = \sigma_{\Delta x}^2 + \sigma_{\lambda x}^2 + \sigma_{\lambda v}^2 \cdot \delta_d^2 + \sigma_{\lambda a}^2 \cdot \delta_d^4,$$

$$D_{\text{чД}}[\Delta\hat{v}_j] = \frac{2\sigma_{\Delta x}^2}{\delta_d^2} + \sigma_{\lambda v}^2 + \sigma_{\lambda a}^2 \cdot \delta_d^2, \quad D_{\text{чД}}[\Delta\hat{a}_j] = \frac{6\sigma_{\Delta x}^2}{\delta_d^4} + \sigma_{\lambda a}^2. \quad (6.29)$$

Умовами ефективного застосування фільтра Калмана є підвищення точності оцінок параметрів руху у порівнянні з методами чисельного диференціювання, тобто виконання співвідношень

$$k_x = \sqrt{\frac{D_{\text{чД}}[\Delta\hat{x}_j]}{D_{\text{ФК}}[\Delta\hat{x}_j]}} > 1, \quad k_v = \sqrt{\frac{D_{\text{чД}}[\Delta\hat{v}_j]}{D_{\text{ФК}}[\Delta\hat{v}_j]}} > 1, \quad k_a = \sqrt{\frac{D_{\text{чД}}[\Delta\hat{a}_j]}{D_{\text{ФК}}[\Delta\hat{a}_j]}} > 1. \quad (6.30)$$

Третя умова в співвідношеннях (6.30) виконується в будь-якому випадку, а перша і друга умови з урахуванням формул (6.28) і (6.29) набувають вигляду:

$$\frac{\sigma_{\Delta x}^2 + \sigma_{\lambda x}^2 + \sigma_{\lambda v}^2 \cdot \delta_d^2 + \sigma_{\lambda a}^2 \cdot \delta_d^4}{\delta_d \sigma_{\Delta x} \sqrt{2\sigma_{\lambda a} \sigma_{\Delta x} + \sigma_{\lambda v}^2}} > 1, \quad \frac{2\sigma_{\Delta x}^2 + \sigma_{\lambda v}^2 \cdot \delta_d^2 + \sigma_{\lambda a}^2 \cdot \delta_d^4}{\delta_d^3 \sigma_{\lambda a} \sqrt{2\sigma_{\lambda a} \sigma_{\Delta x} + \sigma_{\lambda v}^2}} > 1. \quad (6.31)$$

Наприклад, розглянемо рівномірний поступальний або обертальний рух ОБ. Якщо мають місце випадкові похибки підтримання швидкості руху та випадкові коливання, що впливають на поточну координату ОБ в площині зображення, то $\Lambda(t) = [\lambda_x(t) \quad \lambda_v(t) \quad 0]^T$, $\sigma_{\lambda x}^2 \neq 0$, $\sigma_{\lambda v}^2 \neq 0$ і $\sigma_{\lambda a}^2 = 0$. Тоді на основі формул (6.31) при $\delta_d \ll 1$ отримуємо умову ефективного застосування фільтра Калмана для оцінки параметрів руху:

$$\delta_d < \frac{\sigma_{\Delta x}^2 + \sigma_{\lambda x}^2}{\sigma_{\Delta x} \sigma_{\lambda v}}. \quad (6.32)$$

Наприклад, ОБ рухається з постійною швидкістю $v_{cx} = 100$ д.т./с = 100 мм/с в площині відеозображення. Максимальна похибка визначення координат об'єкта $\Delta_{xc} = 0,500$ д.т. = 0,500 мм, її середньоквадратичне значення $\sigma_{\Delta x} = 0,289$ д.т. = 0,289 мм. На об'єкт діють випадкові збурення з максимальною амплітудою $\Delta_{\lambda x} = 10,0$ д.т. = 10,0 мм, їх середньоквадратичне значення $\sigma_{\lambda x} = \Delta_{\lambda x} / 3 = 3,33$ д.т. = 3,33 мм. Випадкові похибки підтримання швидкості руху дорівнюють 5 %, тобто $\Delta_{\lambda v} = 5,00$ д.т./с = 5,00 мм/с, $\sigma_{\lambda v} = \Delta_{\lambda v} / 3 = 1,67$ д.т./с = 1,67 мм/с. Тоді згідно формул (6.28) і (6.29) похибки визначення параметрів руху є такими:

$$\sigma_{\text{ФК}}[\Delta\hat{x}_j] = \sqrt{D_{\text{ФК}}[\Delta\hat{x}_j]} = 0,019 \text{ д.т.} = 0,019 \text{ мм},$$

$$\sigma_{\text{ФК}}[\Delta\hat{v}_j] = \sqrt{D_{\text{ФК}}[\Delta\hat{v}_j]} = 0,$$

$$\sigma_{\text{чД}}[\Delta\hat{x}_j] = \sqrt{D_{\text{чД}}[\Delta\hat{x}_j]} = 3,35 \text{ д.т.} = 3,34 \text{ мм},$$

$$\sigma_{\text{чД}}[\Delta\hat{v}_j] = \sqrt{D_{\text{чД}}[\Delta\hat{v}_j]} = 11,79 \text{ д.т./с} = 10,35 \text{ мм/с}.$$

При цьому умова (6.32) виконується, а коефіцієнт підвищення точності $k_x = 175,9$ разів (при $T_c \rightarrow \infty$).

Фільтр Калмана забезпечує оцінку параметрів руху ОБ в реальному часі. Оцінки параметрів руху обчислюються на основі результатів вимірювань координат ОБ від початку вимірювань до поточного моменту часу. Тому для параметрів руху, що не змінюються в часі, точність оцінки збільшується із збільшенням часу спостереження за ОБ. Але реальні ОБ можуть змінювати параметри свого руху у довільні моменти часу. В цьому випадку виникає значна динамічна похибка оцінок параметрів руху на основі фільтра Калмана.

Для зменшення динамічної похибки потрібно використовувати алгоритмічні процедури, що виконуються в реальному часі і використовують на кожному кроці оцінювання фіксовану кількість вимірюваних значень координат N_{ou} . Наприклад, це можуть бути оцінки на основі цифрових фільтрів з фіксованим об'ємом пам'яті, цифрових фільтрів з експоненційним згладжуванням, оцінки максимальної правдоподібності на основі фіксованої кількості вимірюваних значень координат тощо [27, 28, 33, 45].

Тому розроблено метод алгоритмічної обробки часових послідовностей відеозображень, що забезпечує в реальному часі ідентифікацію змінних значень параметрів руху та визначення поточних координат ОБ. В цьому методі обчислюється оцінка параметрів руху, яка відповідає поточному моменту часу, на основі фіксованої кількості відліків координати ОБ, що виміряні у попередні моменти часу.

При оцінці параметрів руху ідентифікації підлягає вектор [28]:

$$\hat{\beta} = B^{-1}A,$$

$$A_l = \frac{\delta_d^{l-1}}{\sigma_{\Delta x}^2 (l-1)!} \sum_{s=1}^{N_{ou}} s^{l-1} x_{(j-N_{ou}+s)}^*, \quad B_{lk} = \frac{\delta_d^{l+k-2}}{\sigma_{\Delta x}^2 (l-1)!(k-1)!} \sum_{s=1}^{N_{ou}} s^{l+k-2}, \quad (6.33)$$

$$x_i = \beta_1 + \beta_2 t_i + \beta_3 t_i^2 / k_t + \dots,$$

де $\beta_1 = x_0$, $\beta_2 = v_0 = \left. \frac{dx(t)}{dt} \right|_{x=x_0}$, $\beta_3 = a_0 = \left. \frac{d^2 x(t)}{dt^2} \right|_{x=x_0}$ – відповідно початкові

значення координати, швидкості і прискорення ОБ, $j = N_{ou}, \dots, K$, K – загальна кількість результатів вимірювань.

Підраховуючи в формулах (6.33) вирази із змінними l і k , отримуємо:

$$A = \frac{1}{\sigma_{\Delta x}^2} \left[\sum_{s=1}^{N_{ou}} x_{(j-N_{ou}+s)}^* \quad \delta_d \cdot \sum_{s=1}^{N_{ou}} s x_{(j-N_{ou}+s)}^* \quad \frac{\delta_d^2}{2} \cdot \sum_{s=1}^{N_{ou}} s^2 x_{(j-N_{ou}+s)}^* \right]^T, \quad (6.34)$$

$$B = \frac{N_{\text{оц}}}{\sigma_{\Delta x}^2} \begin{bmatrix} 1 & \frac{\delta_d}{2}(N_{\text{оц}}+1) & \frac{\delta_d^2}{12}(N_{\text{оц}}+1)(2N_{\text{оц}}+1) \\ \frac{\delta_d}{2}(N_{\text{оц}}+1) & \frac{\delta_d^2}{6}(N_{\text{оц}}+1)(2N_{\text{оц}}+1) & \frac{\delta_d^3}{8}N_{\text{оц}}(N_{\text{оц}}+1)^2 \\ \frac{\delta_d^2}{12}(N_{\text{оц}}+1)(2N_{\text{оц}}+1) & \frac{\delta_d^3}{8}N_{\text{оц}}(N_{\text{оц}}+1)^2 & \frac{\delta_d^4}{120}(N_{\text{оц}}+1)(2N_{\text{оц}}+1)(3N_{\text{оц}}^2+3N_{\text{оц}}-1) \end{bmatrix},$$

$$\Psi_{\hat{\beta}} = B^{-1} = \frac{\sigma_{\Delta x}^2}{N_{\text{оц}}} \times \begin{bmatrix} \frac{3(3N_{\text{оц}}^2+3N_{\text{оц}}+2)}{(N_{\text{оц}}^2-3N_{\text{оц}}+2)} & -\frac{18(2N_{\text{оц}}+1)}{\delta_d(N_{\text{оц}}^2-3N_{\text{оц}}+2)} & \frac{60}{\delta_d^2(N_{\text{оц}}^2-3N_{\text{оц}}+2)} \\ \frac{18(2N_{\text{оц}}+1)}{\delta_d(N_{\text{оц}}^2-3N_{\text{оц}}+2)} & \frac{12(16N_{\text{оц}}^2+30N_{\text{оц}}+11)}{\delta_d^2(N_{\text{оц}}^4-5N_{\text{оц}}^2+4)} & -\frac{360}{\delta_d^3(N_{\text{оц}}^3-N_{\text{оц}}^2-4N_{\text{оц}}+4)} \\ \frac{60}{\delta_d^2(N_{\text{оц}}^2-3N_{\text{оц}}+2)} & -\frac{360}{\delta_d^3(N_{\text{оц}}^3-N_{\text{оц}}^2-4N_{\text{оц}}+4)} & \frac{720}{\delta_d^4(N_{\text{оц}}^4-5N_{\text{оц}}^2+4)} \end{bmatrix} \quad (6.35)$$

Формули (6.33) – (6.35) забезпечують оцінку параметрів руху, приведену до початку інтервалу оцінювання. Для приведення цієї оцінки до моменту закінчення інтервалу оцінювання і отримання значень параметрів руху в реальному часі необхідно: змінити порядок обчислення сум в формулі (6.34) на протилежний (від закінчення до початку інтервалу оцінювання); змінити знак виразу для швидкості, обчисленого із зміненим порядком сум, на протилежний.

Підставляючи вектор (6.34) і матрицю (6.35) в формулу (6.33) з врахуванням вказаних змін, отримуємо оцінку параметрів руху в реальному часі:

$$\hat{x}_j = \frac{3}{N_{\text{оц}}(N_{\text{оц}}^2-3N_{\text{оц}}+2)} \times \left[(3N_{\text{оц}}^2+3N_{\text{оц}}+2) \cdot \sum_{s=1}^{N_{\text{оц}}} x_{j-s+1}^* - 6(2N_{\text{оц}}+1) \cdot \sum_{s=1}^{N_{\text{оц}}} s x_{j-s+1}^* + 10 \cdot \sum_{s=1}^{N_{\text{оц}}} s^2 x_{j-s+1}^* \right],$$

$$\hat{v}_j = \frac{6}{N_{\text{оц}}\delta_d} \cdot \left[\frac{3(2N_{\text{оц}}+1)}{N_{\text{оц}}^2-3N_{\text{оц}}+2} \cdot \sum_{s=1}^{N_{\text{оц}}} x_{j-s+1}^* - \frac{2(16N_{\text{оц}}^2+30N_{\text{оц}}+11)}{N_{\text{оц}}^4-5N_{\text{оц}}^2+4} \cdot \sum_{s=1}^{N_{\text{оц}}} s x_{j-s+1}^* + \frac{30}{N_{\text{оц}}^3-N_{\text{оц}}^2-4N_{\text{оц}}+4} \cdot \sum_{s=1}^{N_{\text{оц}}} s^2 x_{j-s+1}^* \right], \quad (6.36)$$

$$\hat{a}_j = \frac{60}{N_{\text{оц}} \delta_d^2} \cdot \left[\frac{1}{N_{\text{оц}}^2 - 3N_{\text{оц}} + 2} \cdot \sum_{s=1}^{N_{\text{оц}}} x_{j-s+1}^* - \frac{6}{N_{\text{оц}}^3 - N_{\text{оц}}^2 - 4N_{\text{оц}} + 4} \cdot \sum_{s=1}^{N_{\text{оц}}} s x_{j-s+1}^* + \frac{6}{N_{\text{оц}}^4 - 5N_{\text{оц}}^2 + 4} \cdot \sum_{s=1}^{N_{\text{оц}}} s^2 x_{j-s+1}^* \right]$$

Якщо вважати, що $N_{\text{оц}} \gg 1$, то на основі формул (6.36) отримуємо спрощений вираз для обчислення оцінок параметрів руху в реальному часі:

$$\begin{aligned} \hat{x}_j &= \frac{9}{N_{\text{оц}}} \cdot \sum_{s=1}^{N_{\text{оц}}} x_{j-s+1}^* - \frac{36}{N_{\text{оц}}^2} \cdot \sum_{s=1}^{N_{\text{оц}}} s x_{j-s+1}^* + \frac{30}{N_{\text{оц}}^3} \cdot \sum_{s=1}^{N_{\text{оц}}} s^2 x_{j-s+1}^*, \\ \hat{v}_j &= \frac{36}{N_{\text{оц}}^2 \delta_d} \cdot \sum_{s=1}^{N_{\text{оц}}} x_{j-s+1}^* - \frac{192}{N_{\text{оц}}^3 \delta_d} \cdot \sum_{s=1}^{N_{\text{оц}}} s x_{j-s+1}^* + \frac{180}{N_{\text{оц}}^4 \delta_d} \cdot \sum_{s=1}^{N_{\text{оц}}} s^2 x_{j-s+1}^*, \\ \hat{a}_j &= \frac{60}{N_{\text{оц}}^3 \delta_d^2} \cdot \sum_{s=1}^{N_{\text{оц}}} x_{j-s+1}^* - \frac{360}{N_{\text{оц}}^4 \delta_d^2} \cdot \sum_{s=1}^{N_{\text{оц}}} s x_{j-s+1}^* + \frac{360}{N_{\text{оц}}^5 \delta_d^2} \cdot \sum_{s=1}^{N_{\text{оц}}} s^2 x_{j-s+1}^*. \end{aligned} \quad (6.37)$$

Оцінки (6.36) і (6.37) забезпечують отримання параметрів руху в реальному часі. Також суттєво зменшується динамічна похибка у порівнянні з фільтром Калмана за умови правильного підбору $N_{\text{оц}}$ з врахуванням динамічних властивостей об'єктів, що досліджуються. Для обчислень в реальному часі необхідно перед початком вимірювань обчислити значення постійних коефіцієнтів (за умови $N_{\text{оц}} = \text{const}$, $\delta_d = \text{const}$) перед сумами в формулах (6.36) і (6.37).

Точність оцінок параметрів руху визначається матрицею (6.35). Діагональні елементи цієї матриці дорівнюють дисперсії оцінок координат, швидкості та прискорення ОБ. При $N_{\text{оц}} \gg 1$ отримуємо такі похибки оцінок параметрів руху у реальному часі:

$$\begin{aligned} D_{\text{ММП}}[\Delta \hat{x}_j] &= \Psi_{\hat{\beta}_{11}} = \frac{3(3N_{\text{оц}}^2 + 3N_{\text{оц}} + 2)}{N_{\text{оц}}(N_{\text{оц}}^2 - 3N_{\text{оц}} + 2)} \sigma_{\Delta x}^2 \approx \frac{9}{N_{\text{оц}}} \sigma_{\Delta x}^2, \\ D_{\text{ММП}}[\Delta \hat{v}_j] &= \Psi_{\hat{\beta}_{22}} = \frac{12(16N_{\text{оц}}^2 + 30N_{\text{оц}} + 11)}{N_{\text{оц}}(N_{\text{оц}}^4 - 5N_{\text{оц}}^2 + 4)} \cdot \left(\frac{\sigma_{\Delta x}}{\delta_d} \right)^2 \approx \frac{192}{N_{\text{оц}}^3} \cdot \left(\frac{\sigma_{\Delta x}}{\delta_d} \right)^2, \\ D_{\text{ММП}}[\Delta \hat{a}_j] &= \Psi_{\hat{\beta}_{33}} = \frac{720}{N_{\text{оц}}(N_{\text{оц}}^4 - 5N_{\text{оц}}^2 + 4)} \cdot \left(\frac{\sigma_{\Delta x}}{\delta_d^2} \right)^2 \approx \frac{720}{N_{\text{оц}}^5} \cdot \left(\frac{\sigma_{\Delta x}}{\delta_d^2} \right)^2. \end{aligned} \quad (6.38)$$

Дисперсія похибок оцінок параметрів руху в формулах (6.38) обернено пропорційно залежить від кількості відліків координати та від інтервалу часу між відеозображеннями в часовій послідовності. Тому для підвищення точності вимірювання параметрів руху необхідно:

– збільшувати кількість відліків координати $N_{\text{оц}}$ при фіксованому інтервалі часу $\delta_{\text{д}}$ (до декількох сотень відліків);

– збільшувати інтервал часу $\delta_{\text{д}}$ при фіксованій кількості відліків координати $N_{\text{оц}}$ (до декількох секунд);

– одночасно збільшувати обидва вказані параметри.

Кожен з трьох варіантів підвищення точності вимірювань параметрів руху може бути реалізований за допомогою сучасних пристроїв формування відеозображень.

Порівняємо отримані результати з точністю методу чисельного диференціювання. Для методу чисельного диференціювання дисперсія оцінок параметрів руху на інтервалі $(N_{\text{оц}} - 1)\delta_{\text{д}} = k_q q \delta_{\text{д}}$ дорівнює:

$$\begin{aligned}
 D_{\text{чд}}[\Delta\hat{x}_j] &= \sigma_{\Delta x}^2, \\
 D_{\text{чд}}[\Delta\hat{v}_j] &= \left(\frac{dv_j}{dx_j}\right)^2 \sigma_{\Delta x}^2 + \left(\frac{dv_j}{dx_{j-q}}\right)^2 \sigma_{\Delta x}^2 = \frac{8}{(N_{\text{оц}} - 1)^2} \cdot \left(\frac{\sigma_{\Delta x}}{\delta_{\text{д}}}\right)^2, \\
 D_{\text{чд}}[\Delta\hat{a}_j] &= \left(\frac{da_j}{dx_j}\right)^2 \sigma_{\Delta x}^2 + \left(\frac{da_j}{dx_{j-q}}\right)^2 \sigma_{\Delta x}^2 + \left(\frac{da_j}{dx_{j-2q}}\right)^2 \sigma_{\Delta x}^2 = \frac{96}{(N_{\text{оц}} - 1)^4} \cdot \left(\frac{\sigma_{\Delta x}}{\delta_{\text{д}}^2}\right)^2.
 \end{aligned} \tag{6.39}$$

На основі формул (6.38) і (6.39) отримано порівняльні характеристики точності параметрів оцінок, що обчислюються у реальному часі, у порівнянні з оцінками чисельного диференціювання (при $N_{\text{оц}} \gg 1$):

$$\begin{aligned}
 k_x &= \sqrt{\frac{D_{\text{чд}}[\Delta\hat{x}_j]}{D_{\text{ММП}}[\Delta\hat{x}_j]}} = \sqrt{\frac{N_{\text{оц}}}{9}} \approx 0,333\sqrt{N_{\text{оц}}}, \\
 k_v &= \sqrt{\frac{D_{\text{чд}}[\Delta\hat{v}_j]}{D_{\text{ММП}}[\Delta\hat{v}_j]}} = \sqrt{\frac{N_{\text{оц}}^3}{24(N_{\text{оц}} - 1)^2}} \approx 0,204\sqrt{N_{\text{оц}}}, \\
 k_a &= \sqrt{\frac{D_{\text{чд}}[\Delta\hat{a}_j]}{D_{\text{ММП}}[\Delta\hat{a}_j]}} = \sqrt{\frac{N_{\text{оц}}^5}{7,5(N_{\text{оц}} - 1)^4}} \approx 0,365\sqrt{N_{\text{оц}}}.
 \end{aligned} \tag{6.40}$$

Визначимо точність параметрів руху для часової послідовності відеозображень. Початкові дані для розрахунку похибок: максимальна похибка визначення координат центра мас ОВ $\Delta_{xc} = 0,5$ д.т. = 0,5мм; її середньоквадратичне значення $\sigma_{\Delta x} = 0,289$ д.т. = 0,289 мм; $N_{\text{оц}} = 101$ відлік. Тоді згідно формул (4.46) і (4.48) $\sigma_{\text{ММП}}[\Delta\hat{x}_j] = 0,149$ д.т. = 0,149 мм, $\sigma_{\text{ММП}}[\Delta\hat{v}_j] = 0,171$ д.т./с = 0,171 мм/с, $\sigma_{\text{ММП}}[\Delta\hat{a}_j] = 0,082$ д.т./с² = 0,082 мм/с², $k_x = 3,35$ рази, $k_v = 2,05$ рази, $k_a = 3,67$ рази.

Таким чином, розроблений метод визначення поточних координат і параметрів руху ОВ в реальному часі забезпечує суттєве підвищення

точності вимірювання механічних величин (в 2,1...3,7 рази для 101 відліку координат ОВ згідно формул (6.40)).

Для високоточного вимірювання параметрів руху об'єктів було розроблено пристрій [31] (рис. 6.20), який використовує: цифрову відеокамеру, що формує часову послідовність відеозображень (основний вимірювальний канал); додатковий вимірювальний канал на основі акселерометра та інтеграторів.

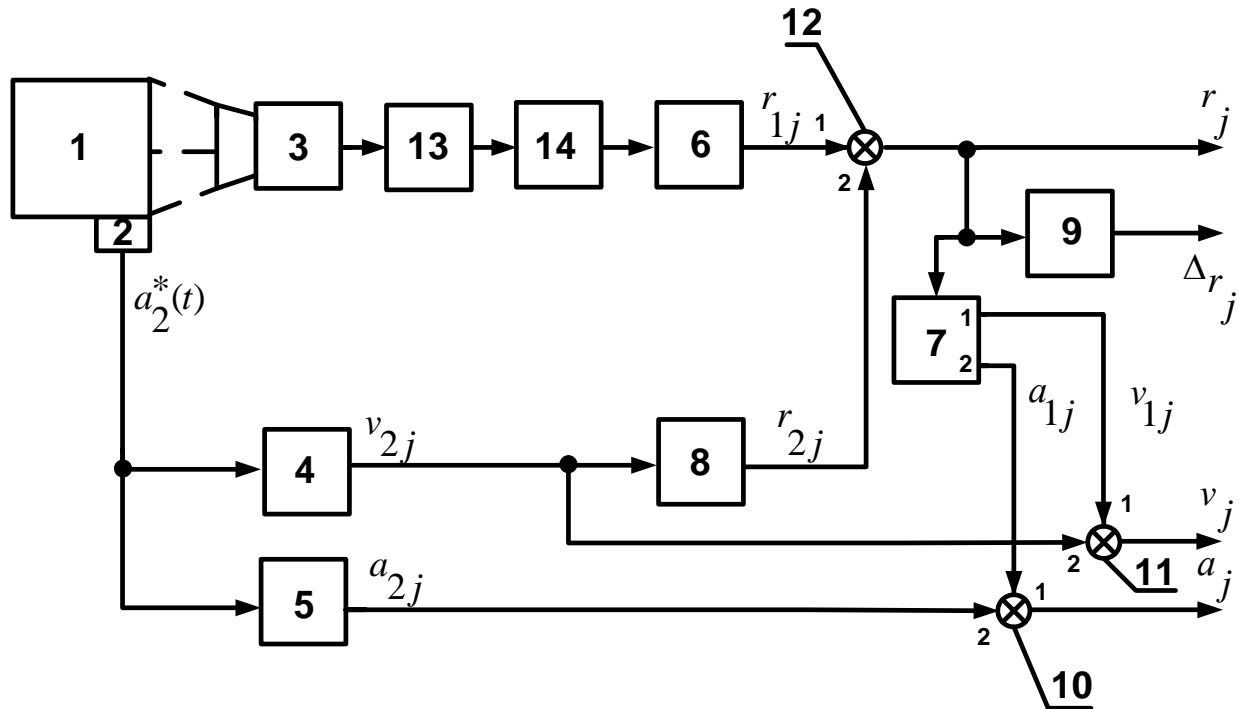


Рис. 6.20. Пристрій для вимірювання параметрів руху об'єктів:

1 – об'єкт вимірювань (ОВ); 2 – акселерометр; 3 - пристрій формування відеозображень (ПФВЗ); 4 - перший інтегратор; 5 - фільтр нижніх частот; 6 - блок визначення поточних координат ОВ; 7 - блок визначення швидкості та прискорення ОВ; 8 - другий інтегратор; 9 - блок обчислення переміщень ОВ; 10 - третій суматор; 11 - другий суматор; 12 - перший суматор

Пристрій для вимірювання параметрів руху об'єктів містить ПФВЗ, аналого-цифровий перетворювач, блок пам'яті, блок визначення поточних координат ОВ, акселерометр, перший та другий інтегратори, фільтр нижніх частот, перший, другий та третій суматори, блок обчислення переміщень ОВ, блок визначення швидкості та прискорення ОВ.

Вхід ПФВЗ оптично пов'язаний з ОВ. Вихід ПФВЗ підключено до входу аналого-цифрового перетворювача, вихід якого підключено до входу блока пам'яті. Вихід блока пам'яті підключено до входу блока визначення поточних координат ОВ, вихід якого підключено до першого входу першого суматора. Вихід цього суматора підключено до входу блока обчислення переміщень ОВ та до входу блока визначення швидкості та прискорення ОВ. Перший вихід цього блока підключено до першого входу другого суматора, а другий вихід підключено до першого входу третього суматора. Вихід

акселерометра підключено до входу фільтра нижніх частот та до входу першого інтегратора, вихід якого підключено до другого входу другого суматора та до входу другого інтегратора. Вихід цього інтегратора підключено до другого входу першого суматора. Вихід фільтра нижніх частот підключено до другого входу третього суматора.

Вихід блока обчислення переміщень ОВ і виходи першого, другого та третього суматорів є виходами всього пристрою для вимірювання параметрів руху об'єктів.

У розробленому пристрої блок пам'яті забезпечує запам'ятовування масиву цифрових даних, що отриманий від аналого-цифрового перетворювача та відповідає всьому відеозображенню ОВ. Блок визначення координат вимірює координати точок, що належать ОВ. При цьому він компенсує похибку, обумовлену нестабільністю зовнішнього освітлення ОВ, за результатами визначення яскравості для всіх точок ОВ на відеозображенні, отриманому від блока пам'яті. Це забезпечує більш повну компенсацію вказаної похибки порівняно з варіантом аналізу яскравості окремих рядків відеозображення та підвищення точності визначення поточних координат і параметрів руху ОВ.

Розроблений пристрій містить акселерометр, що жорстко закріплений на ОВ, та два інтегратори. Така схема забезпечує вимірювання поточних координат ОВ двома незалежними вимірювальними каналами. Перший канал складається з ПФВЗ, аналого-цифрового перетворювача, блока пам'яті та блока визначення поточних координат ОВ і використовує для вимірювань відеозображення цих ОВ. Другий канал складається з акселерометра та двох інтеграторів і використовує для вимірювання поточних координат результат двократного інтегрування вихідного сигналу акселерометра.

Результати вимірювання поточних координат, що отримані у першому та другому вимірювальних каналах, усереднюються першим суматором. При цьому забезпечуються додаткова компенсація похибок вимірювань на основі комплектування двох каналів вимірювання поточних координат ОВ [34, 46].

В результаті зменшується трансформована похибка обчислення переміщень ОВ і підвищується точність пристрою.

Оскільки до блока визначення поточних координат від блока пам'яті надходить все відеозображення ОВ, то блок визначення поточних координат може вимірювати координати як контурних точок ОВ, так і будь-якої іншої точки, що належить цьому об'єкту. В результаті цього забезпечується розширення функціональних можливостей розробленого пристрою.

Введений до розробленого пристрою блок визначення швидкості та прискорення ОВ у сукупності з акселерометром, першим інтегратором та фільтром нижніх частот, другим та третім суматором забезпечує високоточне визначення параметрів руху ОВ і розширення функціональних можливостей розробленого пристрою.

Таким чином, у розробленому пристрої забезпечується підвищення точності та розширення функціональних можливостей вимірювання параметрів руху об'єктів.

Розглянемо роботу розробленого пристрою для вимірювання параметрів руху об'єктів з цифровими відеозображеннями. ПФВЗ формує в моменти часу $t_j = j \cdot \delta_\delta$ часову послідовність відеозображень ОВ, що рухається в просторі відносно ПФВЗ (δ_δ – інтервал часу між сусідніми відеозображеннями у часовій послідовності, $j \in \overline{1, L}$ – номер відеозображення у цій часовій послідовності, L – загальна кількість відеозображень у часовій послідовності). На кожному j -му відеозображенні фіксується поточне положення ОВ. Аналого-цифровий перетворювач перетворює відеозображення в цифрову форму і вони записуються в блок пам'яті. Блок визначення поточних координат ОВ зчитує кожне відеозображення з блока пам'яті, компенсує похибки відеозображень, обумовлені нестабільністю зовнішнього освітлення ОВ та його рухом відносно ПФВЗ, а також визначає поточні координати точок, що належать ОВ на відеозображенні.

При цьому виконується розподіл j -го відеозображення на ОВ і фон шляхом сегментації на основі попередньо заданих значень яскравостей f_ϕ і

f_{ov} для фону і ОВ відповідно, а також порога сегментації $f_n = \frac{f_\phi + f_{ov}}{2}$.

Далі визначаються уточнені значення яскравості ОВ за формулою:

$$\bar{f}_{OVj} = \frac{1}{N_j} \sum_{i=1}^{N_j} f(n_{ij}, m_{ij}),$$

де $f(n_{ij}, m_{ij})$ – яскравість i -ої точки цифрового відеозображення з координатами (n_{ij}, m_{ij}) , що належить ОВ, N – загальна кількість цих точок.

Подібним чином визначається уточнене значення яскравості фону $\bar{f}_{\phi j}$ на j -му цифровому відеозображенні з урахуванням яскравостей точок, що належать фону.

Далі повторюється розподіл j -го відеозображення на ОВ і фон, але з використанням уточненого значення яскравості \bar{f}_{OVj} для ОВ та порога

сегментації $f_{nj} = \frac{\bar{f}_{\phi j} + \bar{f}_{ovj}}{2}$. Така процедура забезпечує повну компенсацію

похибки вимірювання поточних координат ОВ, обумовленої нестабільністю зовнішнього освітлення, та підвищення точності розробленого пристрою.

Для визначення поточних координат контурних точок ОВ використовуються результати розподілу відеозображення на ОВ і фон. Координати інших важливих точок ОВ можуть бути визначені шляхом алгоритмічної обробки координат (n_{ij}, m_{ij}) .

Наприклад, поточні координати центра мас ОВ в моменти часу $t_j = j \cdot \delta_\theta$ визначаються за формулою:

$$x_{1j} = \frac{\delta_x}{N_j} \sum_{i=1}^{N_j} n_{ij}, \quad y_{1j} = \frac{\delta_y}{N_j} \sum_{i=1}^{N_j} m_{ij},$$

де δ_x, δ_y – відстань між сусідніми точками, що належать ОВ на цифровому відеозображенні, по горизонталі і вертикалі відповідно.

Поточне положення центра мас ОВ характеризується радіус-вектором r_{1j} , що поєднує початок координат та центр мас ОВ у площині відеозображення:

$$r_{1j} = \sqrt{x_{1j}^2 + y_{1j}^2}.$$

Акселерометр 2 жорстко закріплений на ОВ і фіксує прискорення, з яким рухається цей об'єкт. Вихідний сигнал $a_2^*(t)$ акселерометра інтегрується першим та другим інтеграторами з урахуванням початкових умов v_0 і r_0 (v_0 – початкова швидкість ОВ, r_0 – його початкова координата). В результаті отримуємо на виході першого інтегратора поточну швидкість ОВ:

$$v_2(t_j) = \int_0^{t_j} a_2(t) dt + v_0.$$

На виході другого інтегратора 8 отримуємо поточне положення ОВ:

$$r_2(t_j) = \int_0^{t_j} v_2(t) dt + r_0.$$

Фільтр 5 нижніх частот забезпечує фільтрацію завад, наявних у вихідному сигналі акселерометра 2, і формує на його виході корисний сигнал прискорення $a_2(t)$ ОВ.

Для подальшого підвищення точності вимірювання параметрів руху у розробленому пристрої потрібно об'єднати результати вимірювань, отримані у першому і другому вимірювальному каналах. Таке об'єднання виконують перший, другий та третій суматори для поточних значень координат, швидкості та прискорення ОВ. Теоретичною основою вказаного об'єднання результатів вимірювань є методи комплексування декількох засобів, що одночасно вимірюють деяку фізичну величину [34, 46]. В результаті на виході першого суматора отримуємо:

$$r_j = \frac{r_{1j} + r_{2j}}{2},$$

де r_{2j} – відліки сигналу $r_2(t_j)$ на виході другого інтегратора, отримані в моменти часу $t_j = j \cdot \delta_\theta$.

Остаточна оцінка поточної координати r_j , що обчислена згідно з наведеною формулою, має підвищену точність та надходить на вхід блока визначення швидкості та прискорення ОВ.

Блок обчислення переміщень розраховує переміщення ОВ:

$$\Delta r_j = r_j - r_{(j-q)},$$

де $q \cdot \delta_\delta$ – інтервал часу, для якого розраховуються переміщення.

Блок визначення швидкості та прискорення розраховує поточні значення швидкості та прискорення ОВ:

$$v_{1j} = \sum_{k=1}^M \eta_v(k) * r_{(j-m+k)}, \quad a_{1j} = \sum_{k=1}^M \eta_a(k) * r_{(j-m+k)},$$

де $\eta_v(k), \eta_a(k)$ – вагові функції, що розраховуються за співвідношеннями:

$$\eta_v(k) = \frac{6}{\delta_\delta M(M^2 - 1)(M^2 - 4)} \cdot [(M + 1)(M + 2)(6M - 7) - 2K(16M^2 - 19) + 30K^2(M - 1)];$$

$$\eta_a(k) = \frac{60}{\delta_\delta^2 M(M^2 - 1)(M^2 - 4)} \cdot [(M + 1)(M + 2) - 6K(M + 1) + 6K^2],$$

M – кількість відеозображень з часової послідовності, що використовується для обчислення поточних значень швидкості і прискорення.

Другий та третій суматори об'єднують результати обчислення швидкості та прискорення, отримані в першому та другому вимірювальному каналі розробленого пристрою:

$$v_j = \frac{v_{1j} + v_{2j}}{2}, \quad a_j = \frac{a_{1j} + a_{2j}}{2},$$

де v_{2j} – відліки сигналу $v_2(t_j)$ на виході першого інтегратора, отримані в моменти часу $t_j = j\delta_\delta$; a_{2j} – відліки сигналу $a_2(t_j)$ на виході фільтра нижніх частот, отримані в моменти часу $t_j = j\delta_\delta$;

Вихідними сигналами розробленого пристрою є поточні значення координати r_j , переміщення Δr_j , швидкості v_j та прискорення a_j , що сформовані на виходах першого суматора, блока обчислення переміщень ОВ, другого та третього суматорів. Ці величини характеризують поточний стан ОВ та його переміщення в момент часу $t_j = j \cdot \delta_\delta$.

Експериментальними дослідженнями на діючому макеті розробленого пристрою встановлено, що точність вимірювання параметрів переміщень у випадку використання двох вимірювальних каналів підвищено у 1,37 рази порівняно з використанням одного вимірювального каналу на базі відеокамери. Таким чином, забезпечено підвищення точності пристрою для вимірювання параметрів руху об'єктів.

6.13. Контрольні питання до розділу 6

1. Дайте загальну характеристику систем автоматичного регулювання (САР).
2. Які найважливіші характеристики об'єктів управління?
3. Назвіть можливі принципи побудови САР.
4. Які існують типи регуляторів?
5. Які можуть бути типи перехідних процесів в САР?
6. Назвіть принципи вибору датчиків для САР.
7. Охарактеризуйте можливості застосування пакету прикладних програм MATLAB для проектування САР.
8. Як вибрати регулятор за його впливом на перехідний процес у САР?
9. Які існують методи синтезу регуляторів?
10. Як оцінити параметри руху об'єктів вимірювань на основі фільтру Калмана?
11. Що таке експоненційне згладжування вимірних значень параметрів об'єктів управління?
12. Як можна обчислити параметри руху об'єктів вимірювань у реальному часі?

РОЗДІЛ 7 ЛАБОРАТОРНИЙ ПРАКТИКУМ З ПРОЕКТУВАННЯ СИСТЕМ УПРАВЛІННЯ

7.1. Лабораторна робота 1. Введення в програму MATLAB математичних моделей систем управління та дослідження їх характеристик

Мета роботи: ознайомитися з робочим простором програми MATLAB, дослідити методи введення математичних моделей систем управління та отримання їх характеристик.

7.1.1. Теоретичні відомості

7.1.1.1. Математичний опис елементів лінійних систем автоматичного керування

Система автоматичного управління (САУ) – сукупність об’єкта управління (ОУ) та управляючого об’єкту (УО).

Реальні системи при теоретичних дослідженнях подають у вигляді моделей, які мають деякий формальний опис, найчастіше математичний.

Математична модель системи – це опис процесів, що проходять в системі, мовою математики.

Для створення моделі САУ необхідно попередньо перевести математичні рівняння, що описують систему, в форму простору стану або знайти передаточні функції системи.

Зв'язок між вхідними і вихідними сигналами лінійної ланки з зосередженими параметрами може бути виражений у вигляді диференційного рівняння:

$$\begin{aligned} a_n \frac{d^n y}{dt^n} + a_{n-1} \frac{d^{n-1} y}{dt^{n-1}} + a_{n-2} \frac{d^{n-2} y}{dt^{n-2}} + \dots + a_1 \frac{dy}{dt} + a_0 y(t) = \\ = b_n \frac{d^n x}{dt^n} + b_{n-1} \frac{d^{n-1} x}{dt^{n-1}} + b_{n-2} \frac{d^{n-2} x}{dt^{n-2}} + \dots + b_1 \frac{dx}{dt} + a_0 x(t). \end{aligned} \quad (7.1.1)$$

В даному випадку $x(t)$ – вхідний сигнал, $y(t)$ – вихідний сигнал, a_i та b_i – сталі коефіцієнти. Таким чином, ланка описується наборами коефіцієнтів $\{a_i\}$ та $\{b_i\}$. Повинна виконуватися нерівність $m \leq n$, тобто максимальний порядок похідної вхідного сигналу не може перевищувати максимального порядку похідної вихідного сигналу. Це пов'язано з неможливістю фізичної реалізації операції “чистого” диференціювання аналогової ланки. Значення n називається *порядком* ланки.

Перетворенням Лапласа називають відношення

$$X(s) = \int_0^{\infty} x(t) e^{-st} dt, \quad (7.1.2)$$

яке ставить функції $x(t)$ дійсної змінної t у відповідність функцію $X(s)$ комплексної змінної $s = \sigma + j\omega$.

При цьому $x(t)$ називається оригіналом, а $X(s)$ – зображенням за Лапласом. Зв'язок між $X(s)$ та $x(t)$ формально може бути записаний у вигляді:

$$X(s) = L[x(t)], \quad (7.1.3)$$

де L – оператор перетворення Лапласа.

Якщо операцію диференціювання замінити на оператор Лапласа $s = \frac{d}{dt}$, то рівняння (7.1.1) зводиться до виразу передаточної функції ланки в операторній формі.:

$$H(s) = \frac{b_m s^m + b_{m-1} s^{m-1} + b_{m-2} s^{m-2} + \dots + b_1 s + b_0}{a_n s^n + a_{n-1} s^{n-1} + a_{n-2} s^{n-2} + \dots + a_1 s + a_0} \quad (7.1.4)$$

В формулі (7.1.4) a_i та b_i – такі самі сталі коефіцієнти, як у наведеному вище диференційному рівнянні (7.1.1).

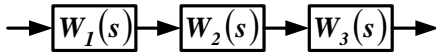
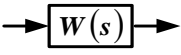
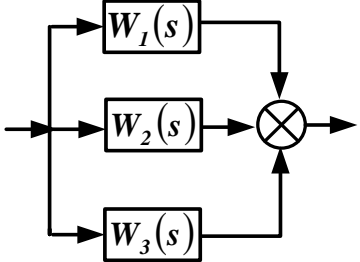
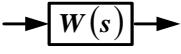
Отже, передаточною функцією ланки (системи) називають відношення зображення за Лапласом вихідної змінної до зображення за Лапласом вхідної змінної при нульових початкових умовах.

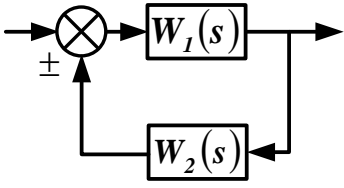
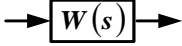
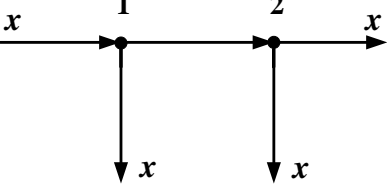
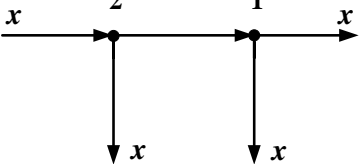
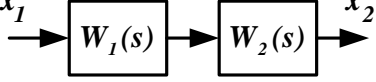
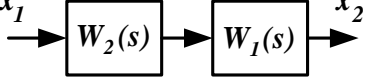
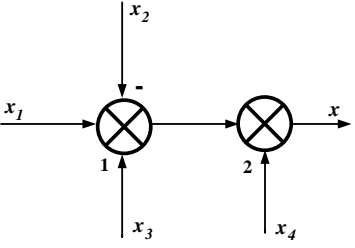
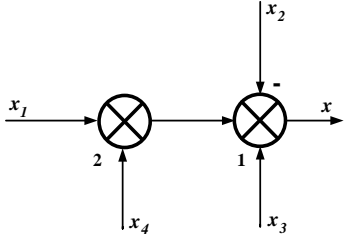
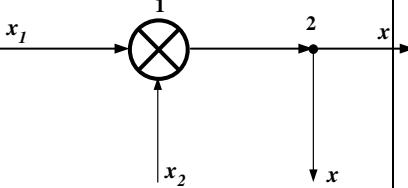
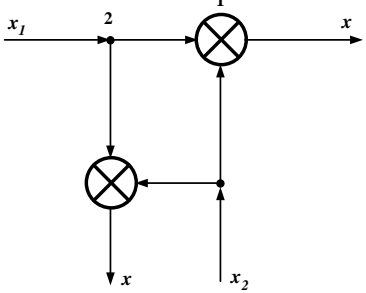
Поняття стійкості системи пов'язане з її здатністю повертатися (з певною точністю) в стан рівноваги після зникнення зовнішніх сил, які вивели її з цього стану. Якщо системи нестійка, то вона не повертається у стан рівноваги, з якого її вивели, вона або віддаляється від нього, або робить навколо нього недопустимі великі коливання.

7.1.1.2. Структурні перетворення лінійних САУ

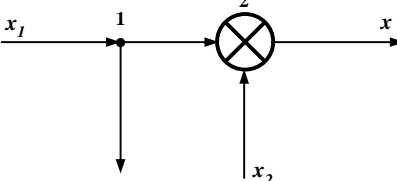
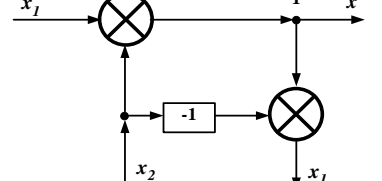
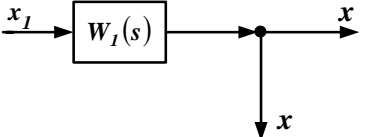
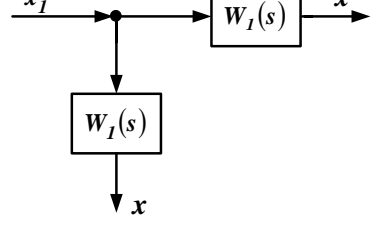
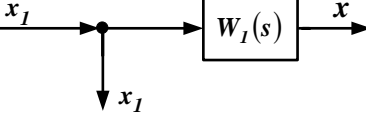
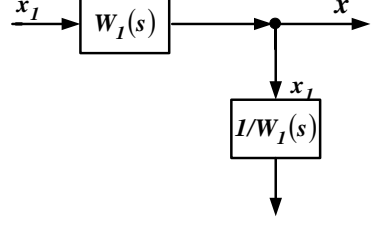
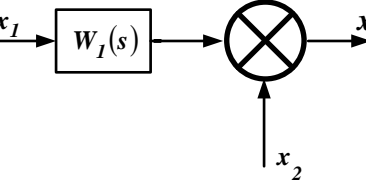
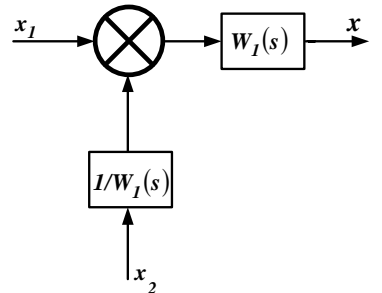
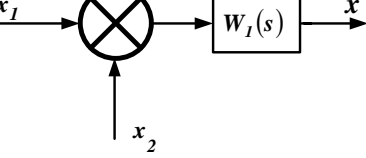
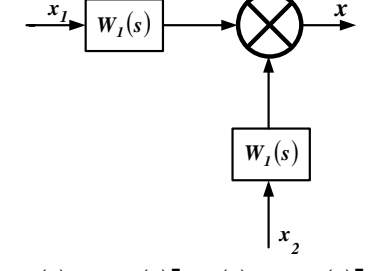
Під час аналізу лінійних САУ часто виникає необхідність у визначенні передаточних функцій як окремих ділянок САУ, так і всієї системи. При цьому користуються основними правилами структурних перетворень, які наведені у таблиці 7.1.1.

Таблиця 7.1.1

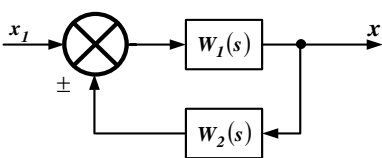
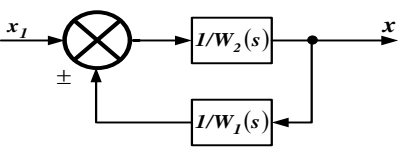
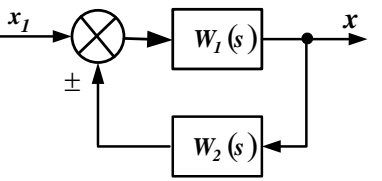
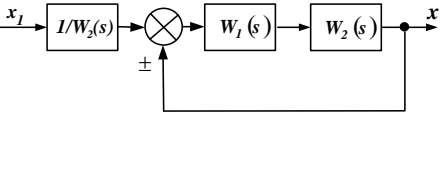
№ п/п	Назва операції	Початкова схема	Еквівалентна схема
1.	Послідовне з'єднання ланок		 $W(s) = W_1(s)W_2(s)W_3(s)$
2.	Паралельне узгоджене з'єднання n ланок		 $W(s) = W_1(s) + W_2(s) + W_3(s)$

1	2	3	4
3.	Паралельне зустрічне з'єднання двох ланок (зворотній зв'язок)		 $W(s) = \frac{W_1(s)}{1 \mp W_1(s)W_2(s)}$
4.	Перестановка вузлів		
5.	Перестановка ланок		
6.	Перестановка суматорів	 $x = x_1 - x_2 + x_3 + x_4$	 $x = x_1 + x_4 - x_2 + x_3$
7.	Перенесення вузла з виходу на вхід суматора	 $x = x_1 + x_2$	 $x = x_1 + x_2$

Продовження таблиці 7.1.1

1	2	3	4
8.	Перенесення вузла з входу на вихід суматора		
9.	Перенесення вузла з виходу на вхід ланки	 $x(s) = W_1(s) X_1(s)$	 $x(s) = W_1(s) X(s)$
10.	Перенесення вузла з входу на вихід ланки		
11.	Перенесення суматора з виходу на вхід ланки	 $x(s) = W_1(s) X_1(s) + X_2(s)$	 $x(s) = W_1(s) X_1(s) + X_2(s)$
12.	Перенесення суматора з входу на вихід ланки	 $x(s) = W_1(s) [X_1(s) + X_2(s)]$	 $x(s) = W_1(s) [X_1(s) + X_2(s)]$

Продовження таблиці 7.1.1

1	2	3	4
13.	Заміна ланок в прямому та зворотному ланцюгах	 $X(s) = \frac{W_1(s)}{1 \mp W_1(s)W_2(s)} X_1(s)$	 $X(s) = \frac{W_1(s)}{1 \mp W_1(s)W_2(s)} X_1(s)$
14.	Заміна ланки в зворотному ланцюзі на одиничний зворотний зв'язок		

Для аналізу одновимірних лінійних САУ користуються трьома видами передаточних функцій: за сигналом управління $W_y(p)$, за похибкою $W_\varepsilon(p)$, за збуренням $W_f(p)$. Розглянемо порядок визначення передаточних функцій, користуючись структурною схемою, що наведена на рис. 7.1.1.

Передаточну функцію за сигналом управління $W_y(p)$ визначають відносно вихідного x та вхідного y сигналів (вихід ε не розглядають, також вважають $f=0$):

$$W_y(p) = \frac{X(p)}{Y(p)} = \frac{W_1(p)W_3(p)}{1 + W_1(p)W_3(p)W_4(p)}. \quad (7.1.5)$$

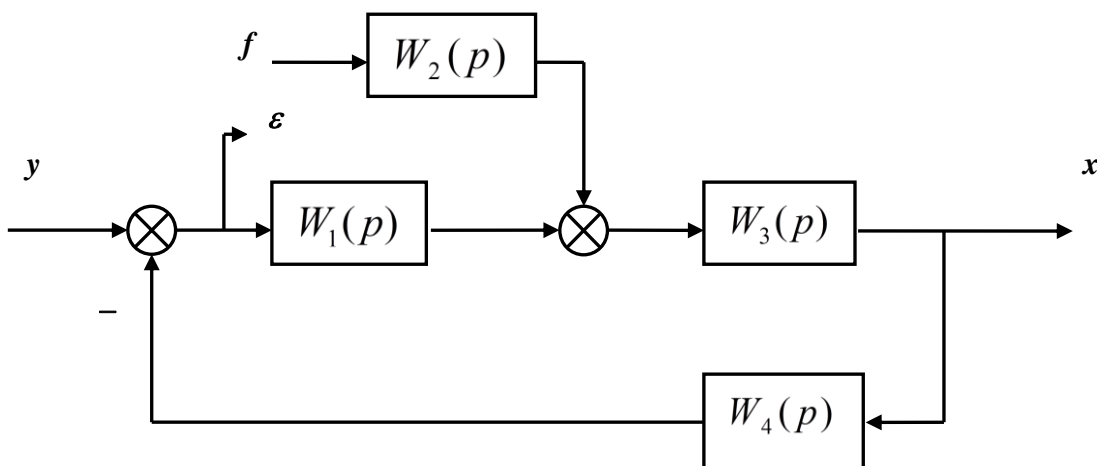


Рис. 7.1.1. Структурна схема САУ, y – вхідний (керуючий сигнал), ε – похибка, f – збурення, x – вихідний сигнал.

Передаточну функцію за похибкою $W_\varepsilon(p)$ визначають відносно похибки ε та входу u (вихід не розглядають, також вважають $f=0$):

$$W_\varepsilon(p) = \frac{\varepsilon(p)}{Y(p)} = \frac{1}{1 + W_1(p)W_3(p)W_4(p)}. \quad (7.1.6)$$

Передаточну функцію за збуренням $W_f(p)$ визначають відносно вихідного сигналу x та збурення f (вихід ε не розглядають, також вважають $u=0$):

$$W_f(p) = \frac{X(p)}{F(p)} = W_2(p) \frac{W_3(p)}{1 + W_3(p)W_4(p)W_1(p)}. \quad (7.1.7)$$

7.1.1.3. Частотні характеристики лінійних САУ

Частотні характеристики лінійних систем (ланок) визначають залежність їх параметрів від частоти. Найчастіше нас буде цікавити:

1. Амплітудна частотна характеристика (**АЧХ**) – залежність відношення амплітуд вихідного та вхідного сигналів від частоти. АЧХ показує, як елемент пропускає сигнали різної частоти (приклад АЧХ наведено на рис. 7.1.2,а);

2. Фазова частотна характеристика (**ФЧХ**) – залежність зсуву фаз між гармонічними вхідними та вихідними системи (ланки). ФЧХ показує, яке запізнення або випередження вихідного сигналу по фазі створює ланка при різних частотах (приклад ФЧХ наведено на рис. 7.1.2,б).

3. Амплітудну і фазову характеристики можна об'єднати в одну загальну - амплітудно-фазову частотну характеристику (**АФЧХ**). АФЧХ являє собою функцію комплексного змінного $j\omega$:

$$W(j\omega) = A(\omega) e^{j\varphi(\omega)} \quad (\text{показникова форма}),$$

де $A(\omega)$ – модуль функції; $\varphi(\omega)$ – аргумент функції.

Кожному фіксованому значенню частоти ω_i відповідає комплексне число $W(j\omega_i)$, яке на комплексній площині можна зобразити вектором, що має довжину $A(\omega_i)$ і кут повороту $\varphi(\omega_i)$ (рис. 7.1.2, в). Від'ємні значення $\varphi(\omega)$, що відповідають відставанню вихідного сигналу від вхідного, прийнято відраховувати за годинниковою стрілкою від позитивного напрямку дійсної осі.

При зміні частоти від нуля до нескінченності вектор $W(j\omega)$ повертається навколо початку координат, при цьому одночасно змінюється довжина вектора. Крива, яку при цьому опише кінець вектора, і є АФЧХ. Кожній точці характеристики відповідає певне значення частоти.

Для аналітичного визначення всіх частотних функцій системи достатньо знайти передаточну функцію на основі перетворення Лапласа $W(p)$. У переважній більшості практичних випадків перехід від $W(p)$ до передатної функції на основі перетворення Фур'є $W(j\omega)$, тобто в частотну область, виконується заміною s на $j\omega$.

$$W(s)|_{s=j\omega} \rightarrow W(j\omega),$$

де: $W(j\omega)$ – амплітудно-фазова частотна характеристики (АФЧХ):

$P(\omega) = \text{Re}[W(j\omega)]$ – дійсна частотна характеристика;

$Q(\omega) = \text{Im}[W(j\omega)]$ – уявна частотна характеристика;

$A(\omega) = \sqrt{P^2(\omega) + Q^2(\omega)}$ - амплітудно-частотна характеристика (АЧХ);

$\varphi(\omega) = \arctg \frac{Q(\omega)}{P(\omega)} \pm k\pi$ - фазочастотна характеристика (ФЧХ), $k=0,1,2,\dots$

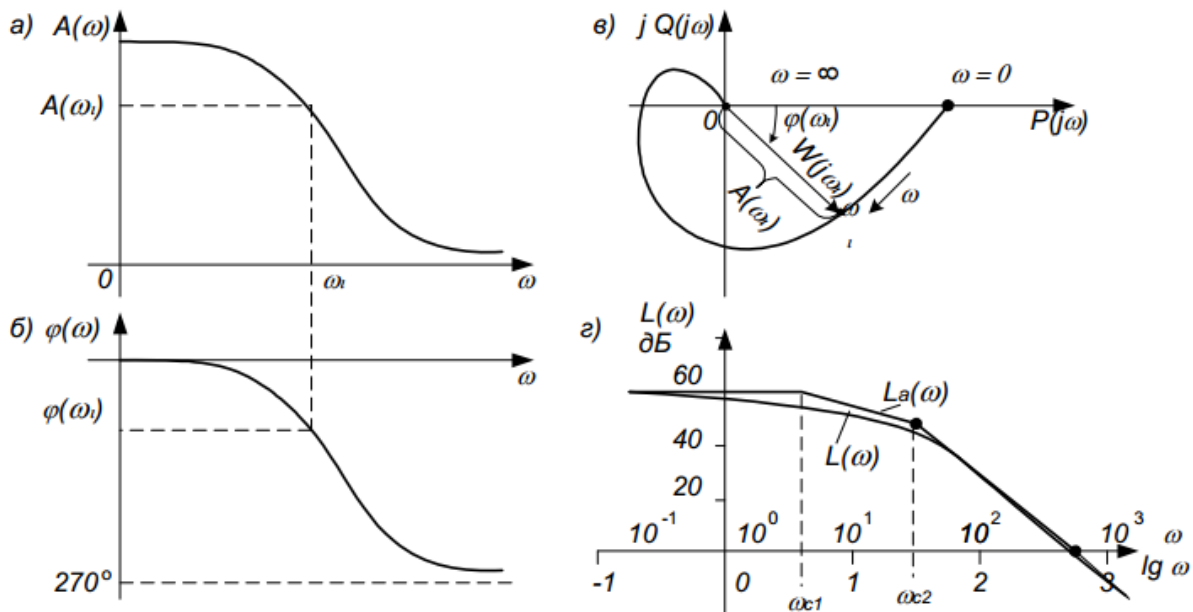


Рис. 7.1.2. Частотні характеристики:

а – амплітудна; б – фазова; в – амплітудно-фазова; г – логарифмічна

На практиці часто використовують логарифмічні частотні характеристики (ЛАХ) – логарифмічну амплітудно-частотну характеристику (ЛАЧХ) та логарифмічну фазочастотну характеристику (ЛФЧХ). Особливість цих характеристик полягає в тому, що по осі абсцис відкладається частота у логарифмічному масштабі, тобто величина $\Omega = \lg \omega$. ЛАЧХ (рис. 7.1.2, г) будують в координатах Ω ; $[L(\omega) = 20 \lg A(\omega)]$, а ЛФЧХ – в координатах Ω ; $\varphi(\omega)$.

7.1.1.4. Часові характеристики лінійних САУ

При проектуванні систем автоматичного управління, крім забезпечення стійкості, доводиться вирішувати проблеми забезпечення потрібних показників якості перехідного процесу (швидкодії, коливальності, перерегулювання, плавності та інших) і точності в усталеному стані.

Розрізняють дві групи показників якості: перша група – показники якості перехідного процесу; друга – показники, що характеризують вимушену (усталену) складову, з якої визначають точність системи.

Показники якості, що визначають безпосередньо за кривою перехідного процесу, називають прямими оцінками якості. У випадках, коли побудова кривої перехідного процесу пов'язана з великими труднощами, використовують непрямі оцінки якості. До непрямих оцінок можна віднести запас стійкості системи по фазі та амплітуді.

Оцінку якості перехідного процесу в системі та її швидкодії можна провести за кривою перехідного процесу при наступних типових вхідних діях: одинична ступінчаста функція $1(t)$, одинична імпульсна функція $\delta(t)$.

До часових характеристик відносять перехідну $h(t)$ і вагову (імпульсну) характеристику $w(t)$.

Перехідною функцією системи називають функцію $h(t)$, що описує зміну вихідної координати системи, коли на її вхід при нульових початкових умовах подається одинична ступінчаста дія.

Графік перехідної функції $h(t)$ від часу t називають перехідною характеристикою.

Імпульсною перехідною або ваговою функцією системи називають функцію, що описує реакцію системи на одиничну імпульсну дію при нульових початкових умовах; позначають цю функцію $w(t)$. Графік імпульсної перехідної функції називають імпульсною перехідною характеристикою.

Зв'язок між часовими характеристиками:

$$w(t) = \frac{dh(t)}{dt}, \quad h(t) = \int_0^t w(t)dt.$$

Існує кілька методів аналітичного і графоаналітичного визначення часових характеристик. Найбільш поширені – метод безпосереднього розв'язування диференціального рівняння та метод зворотного перетворення Лапласа – легко реалізуються на ЕОМ. Якщо в диференційне рівняння підставити в якості вхідного сигналу одиничну ступінчасту функцію $u(t)=1(t)$ і знайти його розв'язок за нульових початкових умов, то одержимо перехідну функцію, тобто $h(t)=x(t)$. Аналогічно, при підстановці $u(t)=\delta(t)$ ($\delta(t)$ – дельта-функція) одержимо $w(t)$. Якщо задана передаточна функція ланки $W(p)$, то часові характеристики найпростіше визначити методом зворотного перетворення Лапласа:

$$[h(t)] = L^{-1}\left[\frac{1}{s}W(p)\right], \quad w(t) = L^{-1}[W(p)]. \quad (7.1.8)$$

Вагова та перехідна функції, як і передаточна функція, є вичерпними характеристиками системи при нульових початкових умовах.

7.1.1.5. Показники якості роботи лінійних САУ

Якість роботи лінійних САУ оцінюють за допомогою кількох видів показників. Використовують прямі, частотні, кореневі, інтегральні показники

якості. Прямі показники якості прийнято визначати за графіком перехідного процесу; такий спосіб є простим, наглядним і має достатню точність. На рис. 7.1.3 представлений типовий коливальний перехідний процес і виконано додаткові побудови для визначення прямих показників якості.

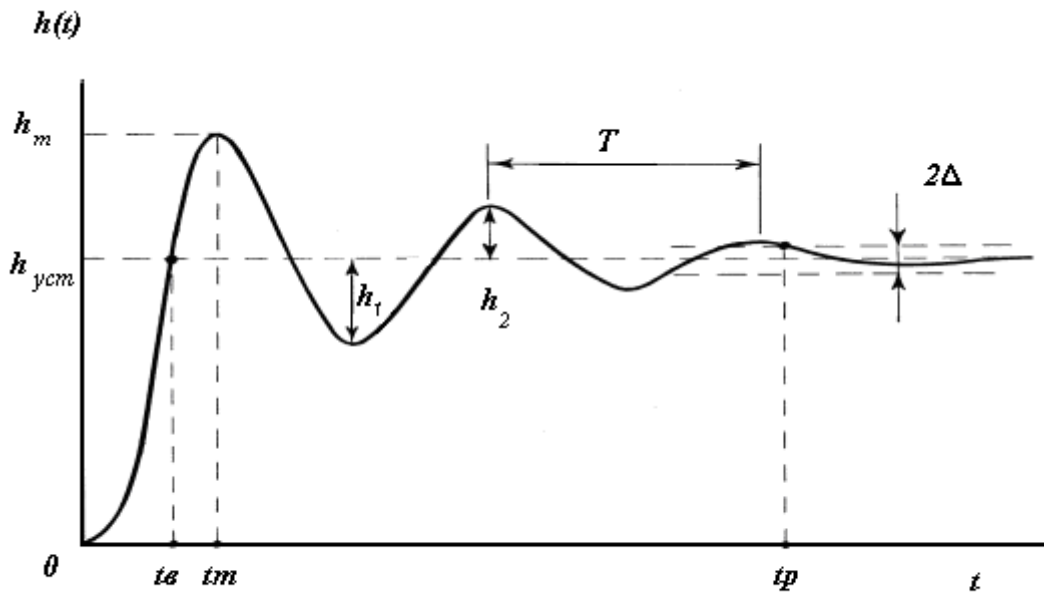


Рис. 7.1.3. Визначення прямих показників якості лінійних САУ

1. Час регулювання (тривалість перехідного процесу) t_p визначається як час від початку процесу до моменту, після якого функція $h(t)$ вже не буде відхилятися від усталеного значення h_{ycm} на величину, більшу ніж задана похибка Δ . Якщо Δ не задана, то її приймають $\Delta = 0,05h_{ycm}$.

2. Перерегулювання σ характеризує максимальне відхилення $h(t)$ від усталеного режиму, що визначається у відсотках:

$$\sigma = \frac{h_m - h_{ycm}}{h_{ycm}} \cdot 100\%. \quad (7.1.9)$$

3. Час виходу на режим t_θ — час від початку перехідного процесу до першого досягнення ним значення h_{ycm} .

4. Час досягнення першого максимуму t_m .

5. Період коливань T_k та відповідно, частота коливань $\omega_k = 2\pi/T_k$

6. Кількість коливань N за час регулювання.

7. Декремент згасання χ характеризує інтенсивність зменшення амплітуди коливань:

$$\chi = \frac{h_1}{h_2}. \quad (7.1.10)$$

Іноді також визначають логарифмічну міру згасання $d = \ln \chi$.

Для перехідних процесів, які не мають усталеного значення ($h(\infty) \rightarrow \infty$), як правило, прямі показники якості не визначають. Якщо $h(0) = 0$ і $h(\infty) \rightarrow 0$, то Δ не визначають, а похибку приймають $\Delta = 0,05h_{\max}$.

7.1.1.6. Методи моделювання систем управління в MATLAB

“Алгебра ланок та кіл”

Реальні системи автоматичного управління складаються із з'єднаних поміж собою окремих блоків (динамічних ланок), рівняння поведінки яких досить прості. Тому в практиці проектування САУ традиційними є структурні методи, коли САУ задається певною схемою з'єднань окремих простих динамічних ланок. В ході проектування необхідно додати до САУ одну або декілька нових ланок таким чином, щоб забезпечити бажану якість динаміки всієї системи. Ці додаткові ланки називаються коригуючими ланками або лінійними регуляторами. Тому в програмі MATLAB передбачена можливість “набору” схеми моделі САУ шляхом попереднього введення моделей ланок, із яких складається система, та наступного з'єднання цих ланок в єдину структуру. Такий метод створення моделі називається “алгебра ланок та кіл”.

До процедур, які виконують з'єднання ланок, відносяться:

- ◆ `plus` (`minus`) – виконує паралельне з'єднання ланок, тобто визначає характеристики моделі системи, що складається з паралельно з'єднаних ланок;
- ◆ `parallel` – виконує паралельне з'єднання ланок. На відміну від попередньої процедури може використовуватися для багатовимірних систем та реалізації паралельного з'єднання лише за деякими входами та виходами;
- ◆ `mtimes` (або знак “*” поміж ланками) – виконує послідовне з'єднання ланок. Використовується лише для одновимірних систем;
- ◆ `series` – послідовне часткове з'єднання багатовимірних систем;
- ◆ `feedback` – з'єднання двох ланок, коли другий елемент складає коло від'ємного зворотного зв'язку для першої ланки;
- ◆ `append` – формальне об'єднання незалежних поміж собою систем (додавання виходів та входів другої системи до виходів та входів першої);
- ◆ `connect` – встановлення з'єднань виходів та входів багатовимірної системи, створеної попередньо процедурою `append`. Схема з'єднань задається матрицею з'єднань Q , що використовується як один із вхідних параметрів процедури;
- ◆ `inv` – створює САК, зворотну даній, тобто таку, в якій виходи та входи переставлені місцями;
- ◆ `vertcat` – виконує так звану вертикальну конкатенацію (зчеплення) систем (ланок), тобто таке їх з'єднання, коли входи цих систем

стають загальними, а виходи залишаються незалежними. Для такого з'єднання необхідно, щоб число входів результуючої системи залишалось таким, як і у кожної із з'єднуємих систем, а число виходів буде дорівнювати сумі числа виходів з'єднуємих систем;

- ◆ `horzcat` – виконує “горизонтальне зчеплення” визначених систем, при якому виходи стають загальними, а входи додаються.

Пакет Control System Toolbox в складі MATLAB має ряд функцій, які виконують аналіз САУ. Це, в першу чергу, функції для отримання відгуків системи на зовнішні впливи, в тому числі відгуків за часом та за частотою. Відгуки за часом визначаються за допомогою таких функцій:

- ◆ `impulse` – обчислення реакції системи на одиничний імпульсний вплив;

- ◆ `step` – обчислення реакції системи на одиничний ступеневий вплив;

- ◆ `initial` – визначення власного руху системи при довільних початкових умовах;

- ◆ `lsim` – обчислення реакції системи на вхідний вплив довільної форми, який задається в формі вектору його значень за часом.

Друга група функцій забезпечує отримання реакції системи в частотній області на зовнішній гармонійний вплив. Це такі функції:

- ◆ `bode` – побудова графіка ЛАХ (діаграми Бодє) системи;

- ◆ `nyquist` – побудова на комплексній площині графіка АФХ системи в полярних координатах;

- ◆ `nichols` – побудова графіка АФХ розімкнутої системи в декартових координатах;

- ◆ `sigma` – побудова графіка АЧХ системи;

- ◆ `margin` – побудова графіка ЛАХ з відображенням запасів за фазою та амплітудою.

Приклад 1.

В САУ послідовно з'єднано дві ланки. Потрібно спростити систему та отримати перехідну, імпульсну, та ЛАХ характеристики.

```
>> w1 = tf(25, [100 50])
```

```
Trasfer function:
```

```
25
```

```
-----
```

```
100 s + 50
```

```
>> w2 = tf(1, [1 0])
```

```
Trasfer function:
```

```
1
```

```
--
```

```
s
```

```
>> W= series (W1, W2)
Transfer function:
      25
-----
100 s^2 + 50 s
```

Визначимо перехідну характеристику:

```
>> step (W) ;
```

Визначимо реакцію системи на одиничний імпульс:

```
>> impulse (W) ;
```

Визначимо графік ЛАХ системи:

```
>> bode (W) ;
```

Створення математичної моделі за умов, що структурна схема спрощена до однієї передаточної функції

Для реалізації цього методу потрібно спростити структурну схему системи автоматичного управління шляхом алгебраїчних перетворень до однієї передаточної функції та ввести результат спрощення в командному рядку MATLAB або у m-файл.

Приклад 2.

Маємо передаточну функцію $W(p) = \frac{1}{p^2 + 0,5p + 1}$. Отримати перехідну характеристику, імпульсну, АЧХ та ЛАХ характеристики САУ.

```
>> sys= tf(1, [1 0.5 1]);
>> step (sys);
>> impulse (sys);
>> sigma (sys);
>> bode (sys);
```

7.1.6.3. Створення математичної моделі САУ в пакеті Simulink

Інструментальний пакет Simulink входить до складу MATLAB та має графічний інтерфейс користувача, за допомогою якого виконується введення структурних схем пристроїв і систем управління. Simulink включає набір бібліотек блоків, необхідних для створення моделей пристроїв і систем управління. Моделі можуть бути ієрархічними, тобто включати підсистеми у вигляді одного блоку.

Розглянемо застосування основних команд і можливостей Simulink, що використовуються при створенні структурної схеми і проведенні моделювання. Даний приклад ілюструє моделювання системи з певною

передаточною функцією та відображення перехідної характеристики замкненої системи (рис. 7.1.4).

Спочатку треба запустити програму MATLAB. Для запуску Simulink необхідно ввести в командному рядку MATLAB команду “Simulink” або знайти на панелі інструментів відповідний значок та запустити його. На екрані з'явиться вікно бібліотек блоків (рис. 7.1.5) і вікно для введення структурної схеми (рис. 7.1.4).

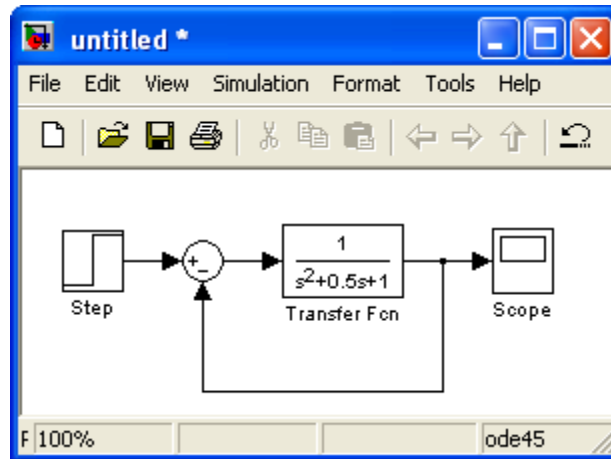


Рис. 7.1.4

Для схеми, що досліджується, необхідно використовувати такі бібліотеки блоків:

- бібліотека джерел сигналів (Sources) - блок ступінчастого сигналу Step;
- бібліотека пристроїв відображення сигналів (Sinks) - блок осцилографа (Scope);

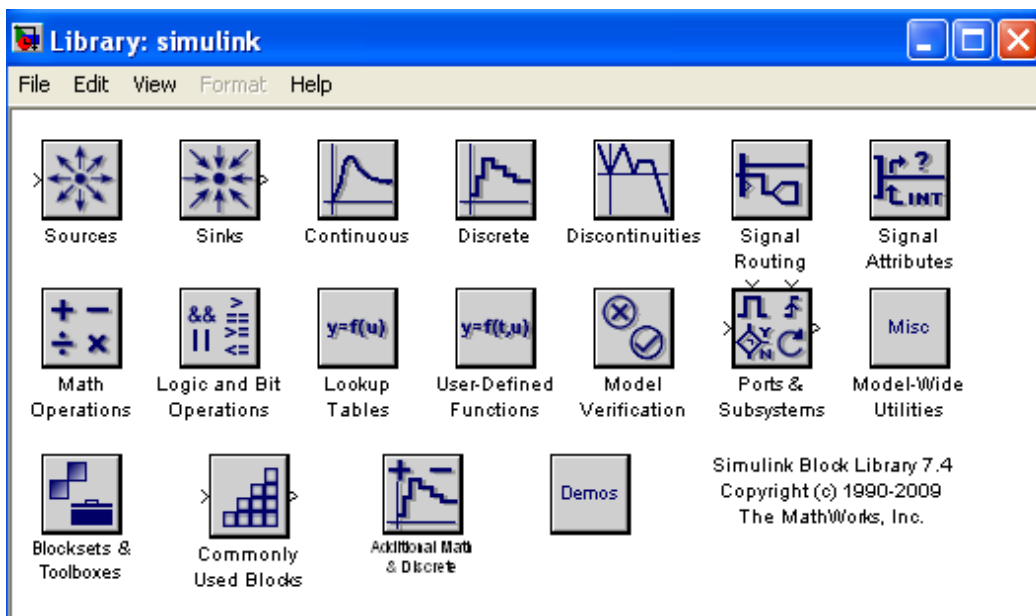


Рис. 7.1.5

– бібліотека лінійних блоків (Continuous) - блок передаточної функції (Transfer Fcn);

– бібліотека блоків математичні операції (Math Operations) - блок суматора (Sum).

Розкрийте вікно бібліотеки джерел сигналів Sources для доступу до блоку ступінчастого сигналу Step. Це виконується подвійним натисненням лівої кнопки миші, коли її курсор знаходиться на піктограмі блоку. Simulink відобразить вікно, що містить усі блоки даної бібліотеки (рис. 7.1.6).

Для додавання блоків в структурну схему їх можна скопіювати з бібліотеки або з іншої моделі. Для копіювання встановіть курсор миші на піктограму потрібного блоку, потім натисніть і утримуйте ліву кнопку миші. Simulink виділить контури блоку і його ім'я. Утримуючи ліву кнопку миші, перемістіть блок у вікно моделі і відпустіть кнопку миші. Копія блоку генератора синусоїдального сигналу Step з'явиться у вікні.

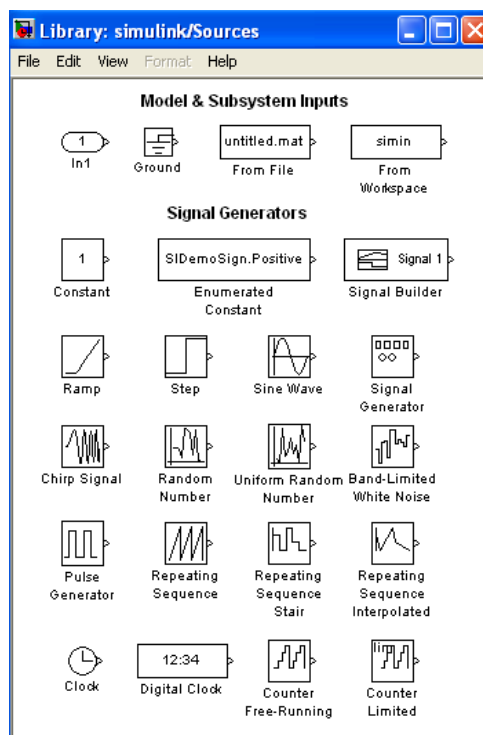


Рис. 7.1.6

Таким же чином введіть інші блоки у вікно структурної схеми. Переміщати блоки в середині вікна можна так же, як і копіювати блоки із бібліотеки. Виділений блок можна переміщати, використовуючи клавіші керування курсором, а також можна використовувати “гарячі” клавіші *CTRL+I*.

Після того, як усі необхідні блоки введені у вікно структурної схеми, вона буде мати такий вигляд, як на рис. 7.1.7.

Кожен блок має декілька символів “>”, що розміщені вздовж його контура. Символ “>”, що вказує напрям в середину блоку, називається

вхідним портом (входом блоку), а символ “>”, що вказує напрям із блоку, називається вихідним портом (виходом блоку). Сигнал передається з виходу одного блоку на вхід іншого блоку за допомогою лінії зв'язку. Якщо на вхід або вихід блоку підключена лінія зв'язку, то символ порту зникає.

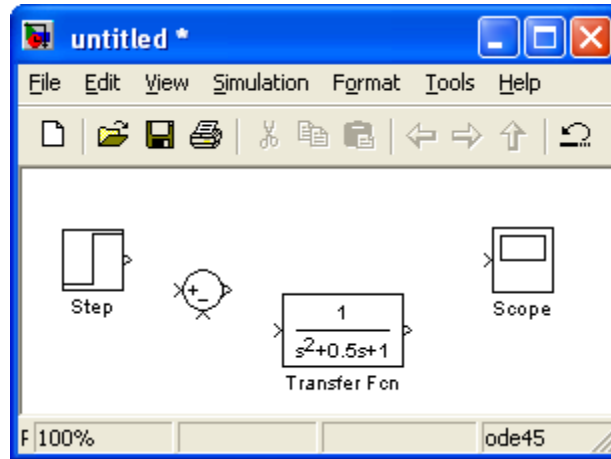


Рис. 7.1.7

Блок суматора Sum має 2 вхідних порти зі знаком «+», а потрібно один зі знаком «+», а інший зі знаком «-» . Для зміни знаку вхідних портів відчинимо вікно параметрів суматора, виконавши подвійне натиснення лівої кнопки миші на даному блоці. У цьому вікні (рис. 7.1.8) змінимо друге значення зі знаком «+» на потрібне нам значення зі знаком «-» і закриємо вікно кнопкою Close. Simulink змінить знаки вхідних портів суматора. Якщо нам знадобиться суматор на 3 і більше портів, аналогічним чином додамо порти з потрібними нам знаками. В даному випадку знаки знаходяться за вертикальною рисою (проміжки між входами), змінюючи знаки перед та за рисою, отримаємо необхідне положення вхідних портів суматора.

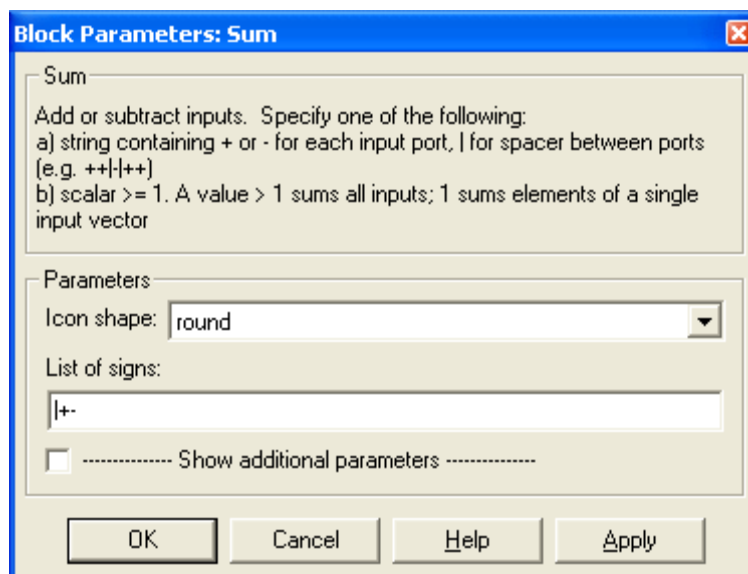


Рис. 7.1.8

З'єднаємо вихід блоку ступінчастого сигналу Step із входом суматора Sum. Встановіть курсор миші на вихідний порт блоку ступінчастого сигналу. Курсор миші змінить свій вигляд на перехрестя. Натисніть і утримуйте ліву кнопку миші і перемістіть курсор миші до вхідного порту суматора. Курсор миші змінить свій вигляд на подвійне перехрестя. Відпустіть ліву кнопку миші. Блоки будуть з'єднані між собою. Блоки також будуть з'єднані, якщо відпустити ліву кнопку миші, коли її курсор знаходиться всередині блоку. Після виконання всіх з'єднань модель буде мати вигляд, як на рис. 7.1.4.

Для моделювання структурної схеми спочатку необхідно відкрити вікно осцилографа Scope подвійним натисненням лівої кнопки миші на його піктограмі. Потім необхідно встановити параметри моделювання. Для цього виконайте команду головного меню Solver – Simulation time і в діалоговому вікні установіть час закінчення моделювання Stop time, що дорівнює 15,0 с.

Запустіть процес моделювання командою Simulation – Start або натискаючи значок «Start» (перевернутий рівнобедрений трикутник). Результати моделювання будуть виведені у вікні Scope.

Моделювання закінчується, коли досягнуто час, зазначений в параметрах моделювання, або якщо виконана команда Simulation – Stop.

Для запису структурної схеми на жорсткий диск виконайте команду File – Save і введіть потрібне ім'я файлу.

7.1.2. Завдання для лабораторної роботи

1. Згідно свого варіанту оберіть структурну схему системи автоматичного управління (додаток 1) та параметри цієї схеми (додаток 2).

2. Ввести дану структурну схему системи автоматичного управління в програму Simulink та отримати перехідну характеристику.

3. Спростити структурну схему системи автоматичного управління шляхом алгебраїчних перетворень до однієї передаточної функції, ввести цю функцію в командному рядку або М-файлі та також отримати перехідну, імпульсну, ЛАЧХ та ЛФЧХ характеристики.

4. Спростити модель даної структурної схеми методом “алгебри ланок та кіл”. Ввести її в командному рядку або М-файлі та також отримати перехідну, імпульсну, логарифмічно-амплітудну характеристики.

5. Оцінити на стійкість систему автоматичного управління за імпульсною, перехідною та за логарифмічно-амплітудною характеристикою.

6. Порівняти графіки характеристик, отримані різними методами моделювання.

7.1.3. Зміст звіту

1. Назва та мета роботи.

2. Структурна схема системи автоматичного управління згідно свого завдання.

3. Результат виконання моделі даної структурної схеми в Simulink та графіки відповідних характеристик.

4. Результат моделювання отриманої передаточної функції в п.2.3 та графіки відповідних характеристик.

4. Результат моделювання методом “алгебри ланок та кіл” та графіки характеристик.

5. Порівняння методів моделювання.

6. Оцінювання на стійкість системи автоматичного управління.

7. Аналіз графіків характеристик, що отримані різними методами моделювання.

8. Висновки по роботі.

7.1.4. Контрольні питання

1. Дайте визначення САУ та її передаточної функції.

2. Дайте визначення часових характеристик та частотних характеристик.

3. Дайте визначення стійкості та показників якості роботи лінійних САУ.

4. Назвіть основні структурні перетворення лінійних САУ.

5. Назвіть основні методи моделювання систем управління в MATLAB.

6. Назвіть процедури, які виконують з'єднання елементів в MATLAB.

7. Назвіть функції, за допомогою яких визначаються відгуки САУ за часом.

8. Назвіть функції, які забезпечують отримання реакції системи в частотній області на зовнішній гармонійний вплив в MATLAB.

9. Які основні бібліотеки блоків використовують для моделювання систем управління в MATLAB?

10. Які блоки ви використовували для моделювання вашої системи управління?

7.2. Лабораторна робота 2. Перетворення математичних моделей систем управління в програмі MATLAB

Мета роботи: ознайомитися з методами перетворення математичних моделей лінійної стаціонарної системи в середовищі програми MATLAB за допомогою інструментальних пакетів Simulink та Control System Toolbox

7.2.1. Теоретичні відомості

7.2.1.1. Введення моделей систем управління різними способами

Введення моделі лінійної САУ до середовища програми MATLAB за допомогою пакетів Simulink та Control System Toolbox можливе в чотирьох форматах:

- 1) у вигляді коефіцієнтів чисельників та знаменників передаточних функцій (поліномів);
- 2) в форматі матриць простору стану;
- 3) в форматі нулів, полюсів та коефіцієнтів передачі системи;
- 4) в форматі доданків простих дробів.

Передаточна функція у вигляді поліномів

Одним з найпростіших форматів є введення моделі у вигляді коефіцієнтів чисельників та знаменників передаточних функцій. Передаточна функція записується у вигляді поліномів чисельника та знаменника.

Даний формат може бути представлений за допомогою пакета Control System Toolbox, наступним чином:

```
>> W=tf([1 -3 2],[1 2 -1 -2]);  
>> impulse(W);  
>>step(W);  
>>bode(W);
```

В результаті отримують імпульсну та перехідну характеристики, а також ЛАЧХ та ЛФЧХ.

Також цей формат математичних моделей може бути представлений за допомогою пакета Simulink (рис. 7.2.1).

В даному випадку на вхід подається одинична ступінчаста дія та одинична імпульсна дія, а на виході відповідно отримують перехідну та імпульсну характеристику.

Складання системи диференційних рівнянь у просторі стану

Як відомо, будь-яка лінійна САУ, в тому числі електромеханічна система, поведінка якої може бути описана звичайним диференційним рівнянням порядку n , завжди може бути подана математичною моделлю у вигляді системи n лінійних диференціальних рівнянь першого порядку:

$$\left. \begin{aligned} \dot{x}_1(t) &= a_{11}x_1(t) + a_{12}x_2(t) + \dots + a_{1n}x_n(t) + b_{11}u_1(t) + b_{12}u_2(t) + \dots + b_{1m}u_m(t); \\ \dot{x}_2(t) &= a_{21}x_1(t) + a_{22}x_2(t) + \dots + a_{2n}x_n(t) + b_{21}u_1(t) + b_{22}u_2(t) + \dots + b_{2m}u_m(t); \\ &\vdots \\ \dot{x}_n(t) &= a_{n1}x_1(t) + a_{n2}x_2(t) + \dots + a_{nn}x_n(t) + b_{n1}u_1(t) + b_{n2}u_2(t) + \dots + b_{nm}u_m(t). \end{aligned} \right\} \quad (7.2.1)$$

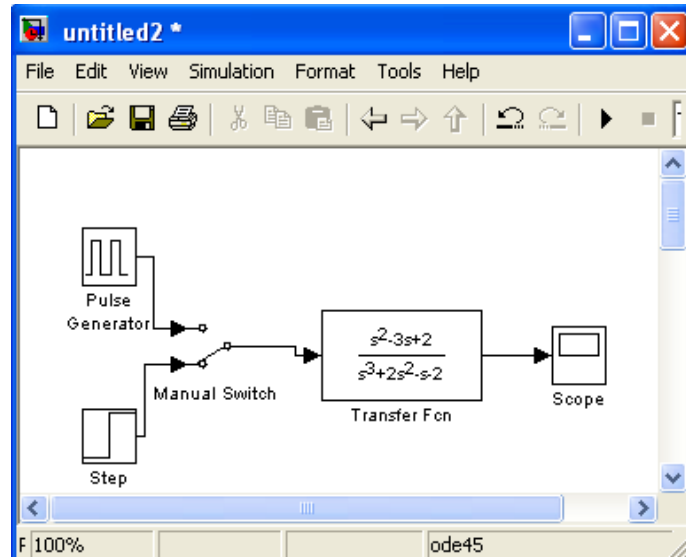


Рис. 7.2.1

Якщо ввести до розгляду матриці коефіцієнтів

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}; \quad B = \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1m} \\ b_{21} & b_{22} & \dots & b_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \dots & b_{nm} \end{bmatrix}. \quad (7.2.2)$$

а також вектори

$$\left. \begin{aligned} X(t) &= [x_1(t) \ x_2(t) \ \dots \ x_n(t)]^T; \\ U(t) &= [u_1(t) \ u_2(t) \ \dots \ u_m(t)]^T, \end{aligned} \right\} \quad (7.2.3)$$

то математичну модель (7.2.1) можна записати у стислій векторно-матричній формі

$$\dot{X}(t) = AX(t) + BU(t), \quad (7.2.4)$$

де $X(t)$ – n - вектор стану системи; $U(t)$ – m - вектор зовнішніх впливів (керувань); A – матриця динаміки системи розміром $n \times n$ (квадратна); B – матриця управління (входу) розміром $n \times m$ (прямокутна).

Модель системи у просторі стану характеризується також рівнянням виходу:

$$Y(t) = CX(t) + DU(t), \quad (7.2.5)$$

де $Y(t)$ – r - вектор виходу системи; C – $(r \times n)$ - матриця відображення динамічних змінних $X(t)$ на вихід системи; D – $(r \times m)$ - матриця компенсації системи (компенсується похибка у вихідному сигналі системи).

Математичні моделі систем у векторно-матричній формі мають дуже важливе практичне значення. Вони широко використовуються в сучасній теорії автоматичного управління при аналітичному конструюванні регуляторів, розробці оптимальних систем управління, тощо. Векторно-матричний опис дозволяє формалізувати процедури розв'язання багатьох складних задач, що дуже важливо при їх розв'язанні за допомогою ЕОМ.

Даний формат математичної моделі САУ може бути представлений за допомогою пакета Control System Toolbox наступним чином:

```
>> W=tf([1 -3 2],[1 2 -1 -2]);
>> [A, B, C, D] = ssdata (W)
A =
    -2.0000    0.5000    1.0000
     2.0000         0         0
         0    1.0000         0
B =
     2
     0
     0
C =
    0.5000   -0.7500    0.5000
D =
     0
>> W1=ss(A,B,C,D);
>> step(W1);
>> impulse(W1);
>>bode(W1);
```

В результаті отримують перехідну та імпульсну характеристики, ЛАЧХ та ЛФЧХ. Також цей формат може бути представлений за допомогою пакета Simulink (рис. 7.2.2).

В даному випадку на вхід подається одинична ступінчаста дія та одинична імпульсна дія, а на виході відповідно отримують перехідну та імпульсну характеристику САУ.

Передаточна функція у вигляді нулів та полюсів

Розклавши чисельник та знаменник передаточної функції (7.1.2) на множники, отримаємо передаточну функцію в наступному вигляді:

$$H(s) = k \frac{(s - z_m)(s - z_{m-1})(s - z_{m-2}) \dots (s - z_1)}{(s - p_n)(s - p_{n-1})(s - p_{n-2}) \dots (s - p_1)}. \quad (7.2.6)$$

В даному випадку $k = \frac{b_m}{a_n}$ - коефіцієнт посилення (gain), z_i - нулі передаточної функції (zero), p_i - полюси передаточної функції (pole). В точках нулів $H(z_i) = 0$, а в точках полюсів $H(p_i) \rightarrow \infty$.

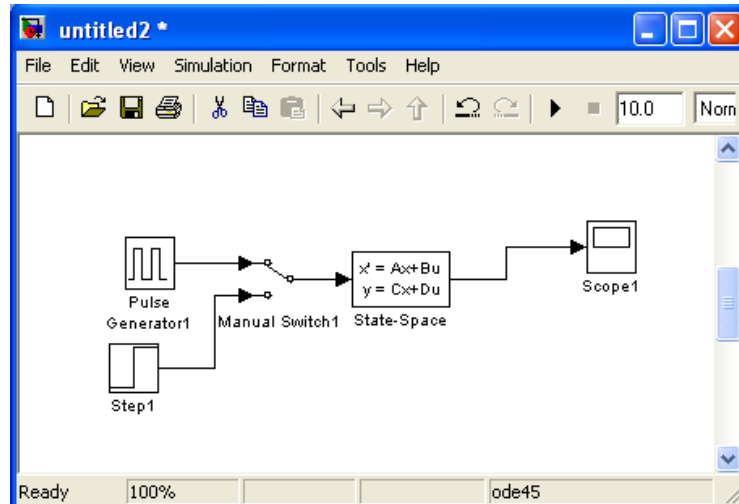


Рис. 7.2.2

Ланка описується набором параметрів $\{z_i\}$, $\{p_i\}$, k .

Нулі передаточної функції можуть бути дійсними або складати комплексно-спряжені пари. Це ж відноситься і до полюсів. Коефіцієнт посилення завжди дійсний.

Даний формат може бути представлений за допомогою пакета Control System Toolbox, наступним чином:

```
>> [z, p, k] = zpkdata (W, 'V')
z =
     2
     1
p =
     1.0000
    -2.0000
    -1.0000
k =
     1
>> W2=zpk(z,p,k)
Zero/pole/gain:
   (s-2) (s-1)
-----
 (s-1) (s+1) (s+2)
>> step(W2)
>> impulse(W2)
>> bode(W2)
```

Також цей формат математичної моделі САУ за допомогою пакета Simulink представлений на рис. 7.2.3.

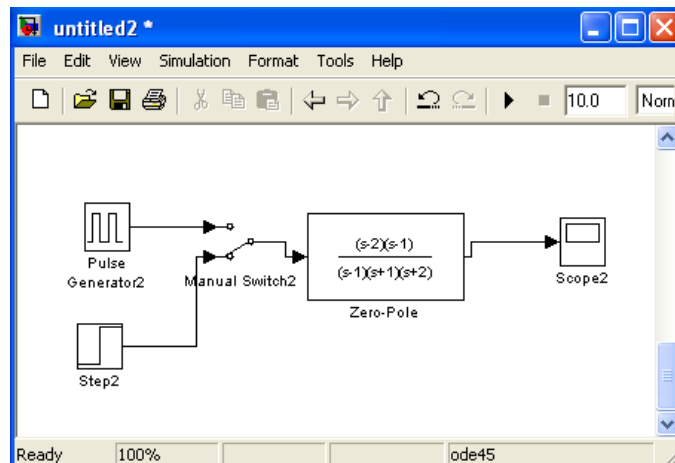


Рис. 7.2.3

В даному випадку на вхід подається одинична ступінчаста дія та одинична імпульсна дія, а на виході відповідно отримують перехідну та імпульсну характеристику.

Передаточна функція у вигляді доданків простих дробів (полюси та лишки)

Ще одним способом перетворення дробово-раціональної передаточної функції (1.4) є її представлення у вигляді суми простих дробів. При відсутності кратних коренів у знаменнику таке представлення має наступний вигляд:

$$H(s) = \frac{r_n}{s - p_n} + \frac{r_{n-1}}{s - p_{n-1}} + \frac{r_{n-2}}{s - p_{n-2}} + \dots + \frac{r_1}{s - p_1} + C_0. \quad (7.2.7)$$

В даному випадку p_i - полюси передаточної функції, числа r_i називаються лишками, C_0 - ціла частина передаточної функції, відмінна від нуля тільки в випадку рівності степенів поліномів чисельника та знаменника.

Ланка описується набором параметрів $\{z_i\}$, $\{p_i\}$, C_0 .

Полюси передаточної функції можуть бути дійсними або складати комплексно-спряжені пари. Лишки, відповідають комплексно-спряженим полюсам, також є комплексно-спряженими.

При наявності кратних полюсів передаточної функції розклад на прості дроби становиться складніше. Кожен m -кратний полюс p_i дає m доданків наступного виду:

$$\frac{r_{i1}}{s - p_i} + \frac{r_{i2}}{(s - p_i)^2} + \frac{r_{i3}}{(s - p_i)^3} + \dots + \frac{r_{im}}{(s - p_i)^m}.$$

Даний форма математичної моделі САУ може бути представлений допомогою пакета Control System Toolbox наступним чином:

```
>> a=[1 -3 2];
>> b=[1 2 -1 -2];
>> [r,p,C]=residue(a,b)
r =
    4.0000
    0.0000
   -3.0000
p =
   -2.0000
    1.0000
   -1.0000
C =
     []

>> [a,b]= residue(r,p,C);
>> W3=tf(a,b);

>> step(W3)
```

Також цей формат математичної моделі САУ за допомогою пакета Simulink представлений на рис. 7.2.4. В даному випадку на вхід подається одинична ступінчаста дія та одинична імпульсна дія, а на виході відповідно отримують перехідну та імпульсну характеристику.

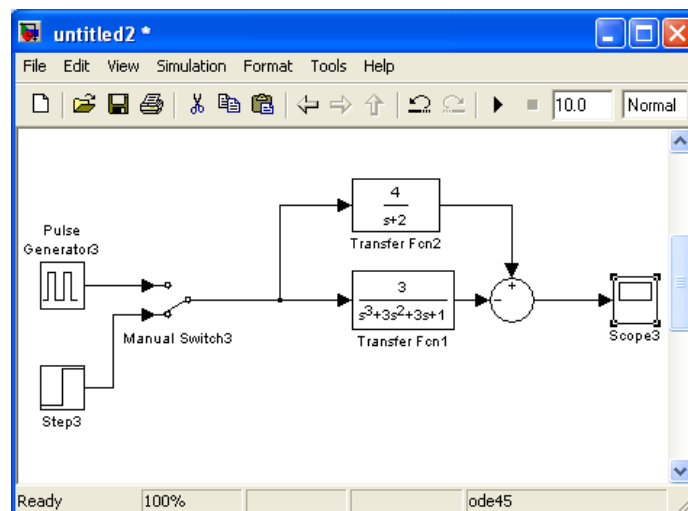


Рис. 7.2.4

7.2.1.2. Введення в програму MATLAB математичних моделей систем управління

Традиційно об'єкти керування або системи описують за допомогою передаточних функцій і тому виникає задача переходу до математичної моделі у формі векторно-матричних диференціальних рівнянь. Такий перехід

від передаточних функцій до простору стану неоднозначний, результат переходу залежить від вектора фазових координат.

До функцій створення LTI-моделей відносяться:

- ◆ `ss` – створює модель простору стану на основі заданих матриць A , B , C , D для рівнянь стану системи (7.2.4) і (7.2.5);
- ◆ `dss` – створює модель простору стану для опису простору стану більш загального вигляду, коли рівняння змінних стану не вирішені відносно похідних;
- ◆ `tf` – створює модель на основі заданих передаточних функцій системи;
- ◆ `zpk` – створює модель на основі заданих нулів, полюсів та коефіцієнтів передачі системи;
- ◆ `filt` – створює модель на основі дискретних передаточних функцій, заданих в формі поліномів змінної z^{-1} ;
- ◆ `set` – задає значення деяких інших полів LTI-моделі (назв входів та виходів, назви системи тощо).

Для отримання окремих характеристик створеної моделі (матриць та векторів, які описують простір стану, коефіцієнтів чисельника та знаменника передаточної функції тощо) можна використовувати такі функції:

- ◆ `tfdata` – отримання векторів чисельника та знаменника передаточної функції системи;
- ◆ `ssdata` – отримання матриць з рівнянь простору стану;
- ◆ `zpkdata` – отримання значень полюсів та нулів системи.

7.2.1.3. Перетворення математичних моделей систем управління

Вище було розглянуто декілька еквівалентних способів математичного опису лінійних ланок. В програмі MATLAB є ряд функцій, призначених для перетворення математичного опису з однієї форми в іншу. Назв цих функцій мають вигляд $xx2yy$, де xx – позначення початкової форми опису, а yy – позначення цільової форми опису ланки.

Необхідність в перетворенні опису часто виникає через те, що функції, призначені для розрахунку ланок, дають результат в одній формі, а функція побудови частотної характеристики САУ потребує задання вхідних параметрів в іншій формі. Далі розглянемо конкретні функції перетворення опису ланок. Для вхідних та вихідних параметрів використовуються наступні позначення:

- 1) передаточна функція:
 - b – вектор-рядок коефіцієнтів (в порядку спадання степенів) чисельника передаточної функції;
 - a – вектор-рядок коефіцієнтів (в порядку спадання степенів) знаменника передаточної функції;
- 2) нулі та полюси:

- z – вектор нулів (стовбець);
- p – вектор полюсів (стовбець);
- k – коефіцієнт підсилення (скляр);

3) простір станів:

- A – квадратна матриця зв'язку вектора стану та його похідної;
- B – вектор-стовбець зв'язку вхідного сигналу та похідної вектора стану;
- C – вектор-рядок зв'язку вихідного сигналу та вектора стану;
- D – скалярний коефіцієнт зв'язку вихідного та вхідного сигналів.

Функція tf2zp перетворює набори коефіцієнтів поліномів чисельника та знаменника передаточної функції в вектори нулів та полюсів, розраховує також значення загального коефіцієнта посилення:

$$[z, p, k] = \text{tf2zp}(b, a);$$

Функція zp2tf є зворотною по відношенню до функції tf2zp: вона здійснює перетворення коефіцієнта посилення, а також векторів нулів та полюсів передаточної функції в коефіцієнти поліномів її чисельника та знаменника:

$$[b, a] = \text{zp2tf}(z, p, k);$$

Функція ss2tf є зворотною по відношенню к функції zp2tf: вона перетворює параметри простору стану в коефіцієнти поліномів передаточної функції ланок:

$$[b, a] = \text{ss2tf}(A, B, C, D);$$

Функція **zp2ss** перетворює нулі, полюси та коефіцієнти підсилення ланки в її параметри простору стану:

$$[A, B, C, D] = \text{zp2ss}(z, p, k);$$

Функція ss2zp є зворотною по відношенню к функції zp2ss, перетворює параметри простору стану в нулі, полюси та коефіцієнти підсилення ланки:

$$[z, p, k] = \text{ss2zp}(A, B, C, D);$$

Функція residue перетворює передаточну функцію, що задана в вигляді коефіцієнтів поліномів чисельника та знаменника, в доданки простих дробів. Вона ж виконує зворотне перетворення. Ця функція відноситься до базової бібліотеки MATLAB.

При двох вхідних параметрах виконується розклад передаточної функції на прості дроби:

$$[r, p, k] = \text{residue}(b, a);$$

В даному випадку b та a – коефіцієнти поліномів чисельника та знаменника передаточної функції відповідно. Вихідні параметри – вектори-стовпці полюсів (p) та відповідні їм лишків (r), а також коефіцієнт підсилення k .

При використанні трьох вхідних параметрів функції residue виконується перетворення лишків, полюсів ті коефіцієнти цілої частини в

коефіцієнти чисельника та знаменника передаточної функції, тобто виконується підсумовування простих дробів:

$$[b, a] = \text{residue}(r, p, k).$$

При перетвореннях ЛТІ-моделей необхідно мати на увазі наступне:

1. Три форми існування ЛТІ-моделей не еквівалентні при чисельних розрахунках. Точність обчислень з передаточними функціями високих порядків може бути незадовільною. Необхідно працювати переважно зі збалансованими моделями простору стану, а передаточні функції використовувати лише для відображення результатів моделювання.

2. Перетворення до формату передаточних функцій може супроводжуватися втратами точності. В результаті, полюси передаточної функції можуть помітно відрізнятись від полюсів початкової ZPK-моделі або моделі простору стану.

3. Перетворення в простір стану є неоднозначним у випадку одновимірної системи та не гарантують створення мінімальної конфігурації системи у випадку багатовимірної системи. Задана в просторі стану модель sys при перетвореннях $ss(tf(sys))$ може сформувати модель з іншими матрицями простору стану або навіть з іншим числом змінних стану в багатовимірному випадку. Таким чином, необхідно, по можливості, уникати зайвого перетворення математичної моделі з одного формату в інший.

7.2.2. Завдання для лабораторної роботи

1. Згідно свого варіанту оберіть структурну схему системи автоматичного управління (Додаток А) та параметри цієї схеми (додаток Б).

2. Введіть передаточну функцію у вигляді поліномів в Control System Toolbox та отримайте перехідну, імпульсну, ЛАЧХ та ЛФЧХ характеристики.

3. Введіть передаточну функцію у вигляді поліномів за допомогою пакету Simulink та отримайте перехідну, імпульсну характеристики.

4. Введіть передаточну функцію у вигляді простору стану в Control System Toolbox та отримайте перехідну, імпульсну, ЛАЧХ та ЛФЧХ характеристики.

5. Введіть передаточну функцію у вигляді простору стану в пакет Simulink та отримайте перехідну, імпульсну характеристики.

6. Введіть передаточну функцію в вигляді нулів та полюсів в Control System Toolbox та отримайте перехідну, імпульсну, ЛАЧХ та ЛФЧХ характеристики.

7. Введіть передаточну функцію у вигляді нулів та полюсів в пакет Simulink та отримайте перехідну, імпульсну характеристики.

8. Введіть передаточну функцію у вигляді простих дробів в Control System Toolbox та отримайте перехідну, імпульсну, ЛАЧХ та ЛФЧХ характеристики.

9. Оцініть стійкість системи автоматичного управління за перехідною характеристикою та за ЛАЧХ, ЛФЧХ.

10. Введіть передаточну функцію у вигляді простих дробів у пакет Simulink та отримайте перехідну, імпульсну характеристики.

11. Порівняйте графіки характеристик, отримані за різними методами представлення передаточної функції.

7.2.3. Зміст звіту

1. Назва та мета роботи.
2. Структурна схема системи автоматичного управління згідно свого варіанту завдання.
3. Результат моделювання САУ в Control System Toolbox та графіки її характеристик.
4. Результат моделювання САУ в Simulink та графіки її характеристик.
5. Порівняння методів представлення передаточної функції.
6. Оцінювання на стійкість системи автоматичного управління.
7. Аналіз графіків характеристик, що отримані різними методами моделювання.
8. Висновки по роботі.

7.2.4. Контрольні питання

1. Як ввести передаточну функцію у вигляді поліномів у пакети Simulink та Control System Toolbox?
2. Як ввести передаточну функцію у вигляді простору стану у пакети Simulink та Control System Toolbox?
3. Як ввести передаточну функцію у вигляді нулів та полюсів у пакети Simulink та Control System Toolbox?
4. Як ввести передаточну функцію у вигляді простих дробів у пакети Simulink та Control System Toolbox?
5. За допомогою яких команд передаточну функцію САУ можна перетворити з одного вигляду в інший?
6. За допомогою яких команд можна отримати перехідну, імпульсну, ЛАЧХ та ЛФЧХ характеристики?
7. Як впливає на моделювання САУ перетворення її моделі із одного формату в інший?

7.3. Лабораторна робота 3. Синтез та моделювання роботи регулятора для системи управління

Мета роботи: дослідити методи вибору регуляторів в середовищі програми Matlab (інструментальні пакети Control System Toolbox та Simulink), поліпшити показників якості перехідних процесів системи автоматичного управління.

7.3.1. Теоретичні відомості

7.3.1.1. Загальні відомості

Пропорційно-інтегрально-диференціальний (ПІД) регулятор – пристрій в керуючому контурі САУ зі зворотним зв'язком. Використовується в системах автоматичного управління для формування керуючого сигналу з метою отримання необхідних точності і якості перехідного процесу. ПІД-регулятор формує управляючий сигнал, який є сумою трьох доданків, перший з яких пропорційний різниці вхідного сигналу і сигналу зворотного зв'язку (сигнал неузгодженості), другий – інтеграл від сигналу неузгодженості, третій – похідна від сигналу неузгодженості.

В даній лабораторній роботі вибір структури та параметрів регуляторів виконується на основі аналізу перехідної характеристики *замкнутої системи*. Така характеристика обчислюється для кожного варіанту побудови регулятора методом математичного моделювання за допомогою програми MATLAB.

Розглянемо характеристики П-, І- та Д-регуляторів (складових частин ПІД-регулятора) і ознайомимося з методами отримання потрібного відгуку системи. Ці методи включають вибір певного типу регулятора. Система з одиничним зворотним зв'язком (система основного типу) зображена на рис. 7.3.1.

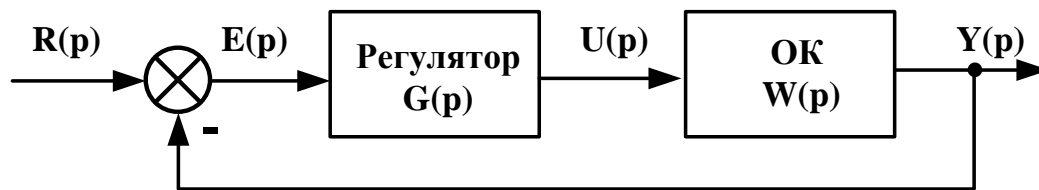


Рис. 7.3.1

Об'єкт управління (ОУ) представляє собою систему, яка має керуватися певним регулятором. Регулятор забезпечує формує вхідний сигнал керування та керує поведінкою всієї системи. В повному обсязі регулятор складається з трьох елементів:

- пропорційного підсилювача з масштабуючим коефіцієнтом k_p ;
- інтегратора з коефіцієнтом перетворення k_i ;
- обчислювача похідної з коефіцієнтом перетворення k_d .

Передаточна функція регулятора в повному складі дорівнює:

$$W(s) = k_p + \frac{k_i}{s} + k_d s = \frac{k_d s^2 + k_p s + k_i}{s}.$$

Проаналізуємо роботу ПД-регулятора в замкнутому колі системи, зображеної на рис. 7.3.1. Похибка на виході суматора $e(t)$ (різниця між вхідним впливом $r(t)$ та вихідним сигналом $y(t)$) надходить до ПД-регулятора, який її масштабує, обчислює похідну та інтеграл від неї і таким чином утворює сигнал керування $u(t)$:

$$u(t) = k_p e(t) + k_i \int e(t) dt + k_d \frac{de(t)}{dt}.$$

Сигнал управління надходить до об'єкта керування, де під впливом цього сигналу створюється нове значення вихідного сигналу $y(t)$. Це нове значення надходить до суматора, де створюється нове значення сигналу похибки $e(t)$. Регулятор отримує це нове значення сигналу похибки і виконує нові обчислення для формування нового значення сигналу управління. Така робота в замкнутому колі буде проходити до того часу, коли зникне різниця між вхідним та вихідним сигналами.

Вплив кожного із елементів регулятора на якість перехідних процесів можна відобразити за допомогою таблиці 7.3.1.

Таблиця 7.3.1

Коефіцієнт регулятора	Час досягнення максимуму	Перерегулювання	Час регулювання	Похибка сталого режиму
k_p	Зменшується	Збільшується	Невеликі зміни	Зменшується
k_i	Зменшується	Збільшується	Збільшується	Ліквідується повністю
k_d	Невеликі зміни	Зменшується	Зменшується	Невеликі зміни

Необхідно відмітити, що наведені в таблиці властивості перехідних характеристик не можуть бути зовсім точно відтворені, тому що складові елементи регулятора впливають один на одного. Фактично зміни параметрів одного елемента можуть обумовлювати зміну дії двох інших елементів. Тому таблицю слід використовувати тільки для попередніх розрахунків при синтезі регулятора.

7.3.1.2. Задача синтезу

Розглянемо просту схему об'єкта керування: деяка маса m утримується пружиною p з коефіцієнтом жорсткості k та демпфуючим пристроєм з коефіцієнтом демпфування b . Рівняння математичної моделі об'єкта записується як рівняння рівноваги:

$$m \frac{d^2x}{dt^2} + b \frac{dx}{dt} + kx = f, \quad (7.3.1)$$

де f – зовнішня сила, що діє на об'єкт;

x – поточна горизонтальна координата об'єкта.

Перетворимо (7.3.1) за Лапласом, що дає:

$$ms^2 X(s) + bsX(s) + kX(s) = F(s). \quad (7.3.2)$$

На основі (7.3.2) запишемо передаточну функцію відносно поточної координати $X(s)$ (вихід об'єкта) та сили $F(s)$ (вхід об'єкта):

$$\frac{X(s)}{F(s)} = \frac{1}{ms^2 + bs + k}. \quad (7.3.3)$$

Будемо вважати, що $m = 1$ кг; $b = 10$ Н·с/м; $k = 20$ Н/м; $f = 1$ Н. Тоді отримуємо

$$\frac{X(s)}{F(s)} = \frac{1}{s^2 + 10s + 20}.$$

При вирішенні задачі синтезу САУ необхідно показати, що кожний із типів регуляторів (пропорційний, інтегральний та диференційний) сприяє:

- підвищенню швидкодії системи;
- мінімізації перерегулювання ;
- зменшенню похибки сталого режиму.

7.3.1.3. Реакція моделі розімкнутої системи на ступеневий сигнал

Для визначення реакції моделі розімкнутої системи на ступеневий сигнал необхідно створити новий m-файл такого змісту:

```
>> W=tf(1, [1 10 20]);
>> step(W)
```

В результаті виконання команд з цього файлу отримуємо графік реакції розімкнутої системи на одиничну ступінчасту функцію – перехідну характеристику (рис. 7.3.2).

Коефіцієнт підсилення в передаточній функції об'єкту дорівнює $1/20$, тому $0,05$ – це стале значення виходу об'єкта. Це означає, що похибка сталого режиму дорівнює $\frac{K}{1+K} = \frac{20}{1+20} = 0,95$ і є неприпустимо великою.

Час досягнення сталого режиму складає близько однієї секунди, а час досягнення на виході величини $0,025$ становить $0,5$ с.

Таким чином отримані параметри перехідного процесу є незадовільними.

Тому постає задача синтезу регулятора, який зменшить час встановлення, а також ліквідує похибку сталого режиму або зменшить її до припустимої величини.

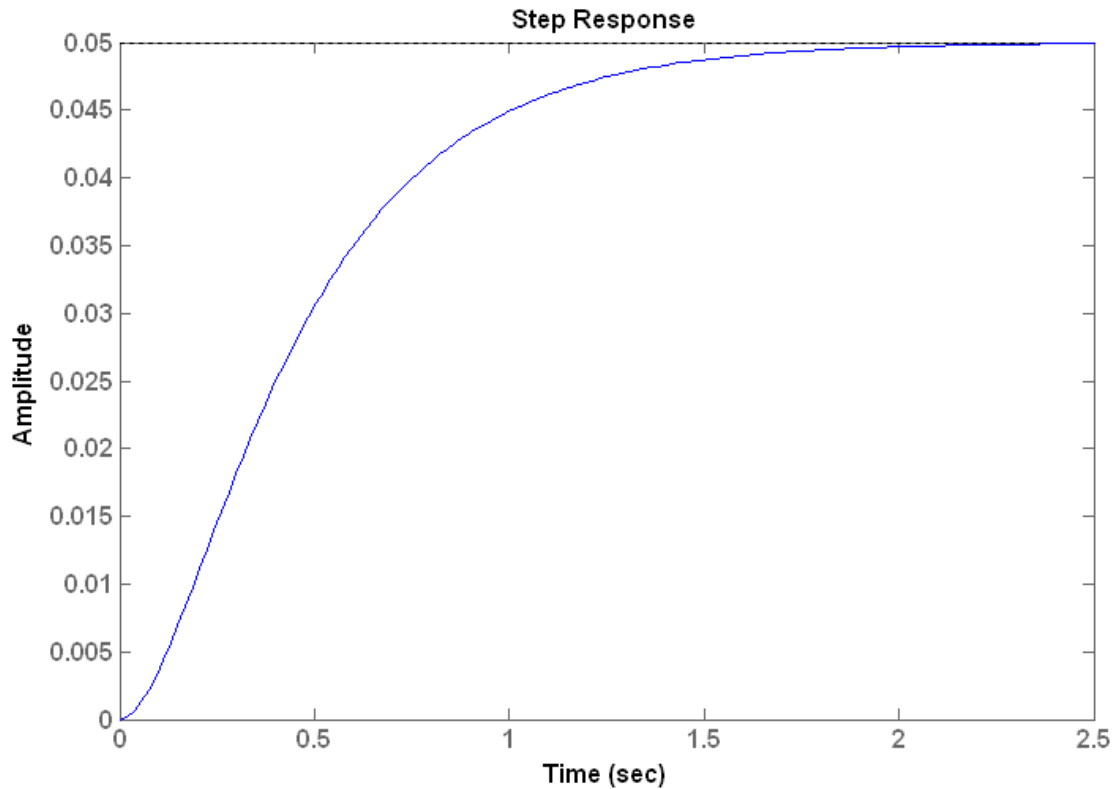


Рис. 7.3.2. Перехідна характеристика початкової САУ

7.3.1.4. Реалізація ПІД-регуляторів за допомогою інструментальних пакетів Simulink та Control System Toolbox

За допомогою інструментального пакету Simulink реалізація ПІД-регуляторів для об'єкту керування з передаточною функцією

$$W(s) = \frac{1}{s^2 + 10s + 20}$$

проводиться наступним чином (рис. 7.3.3):

Недоліком такого методу є те, що оптимальні коефіцієнти регуляторів необхідно знати заздалегідь або використовувати функцію `pidtool`, точність розрахунку якої залежить від точності розрахунку коефіцієнтів в MATLAB.

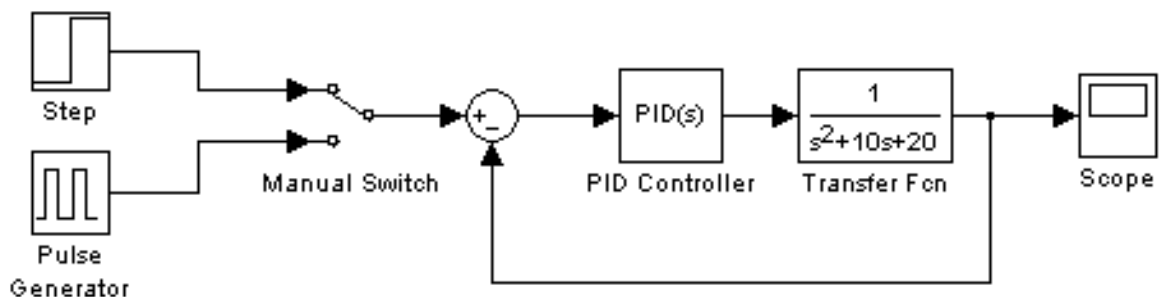


Рис. 7.3.3

За допомогою інструментального пакету Control System Toolbox реалізація ПІД-регуляторів проводиться наступним чином, (рис. 7.3.4):

```
>> W=tf(1,[1 10 20]);
>> pidtool(W)
```

Перевагою такого методу є те, що вид та коефіцієнти регулятора підбираються автоматично. Недоліком методу є те, що точність підбору коефіцієнтів залежить від точності розрахунку в MATLAB результатів функції pidtool.

Також, крім функції pidtool, в MATLAB є функція pidtune, вона розглянута нижче. Перевагою такого методу є те, що вид та коефіцієнти регулятора підбираються автоматично. Недоліком методу є те, що точність підбору коефіцієнтів залежить від точності розрахунку в MATLAB функції pidtune. Також потрібно вручну будувати часові та передаточні функції.

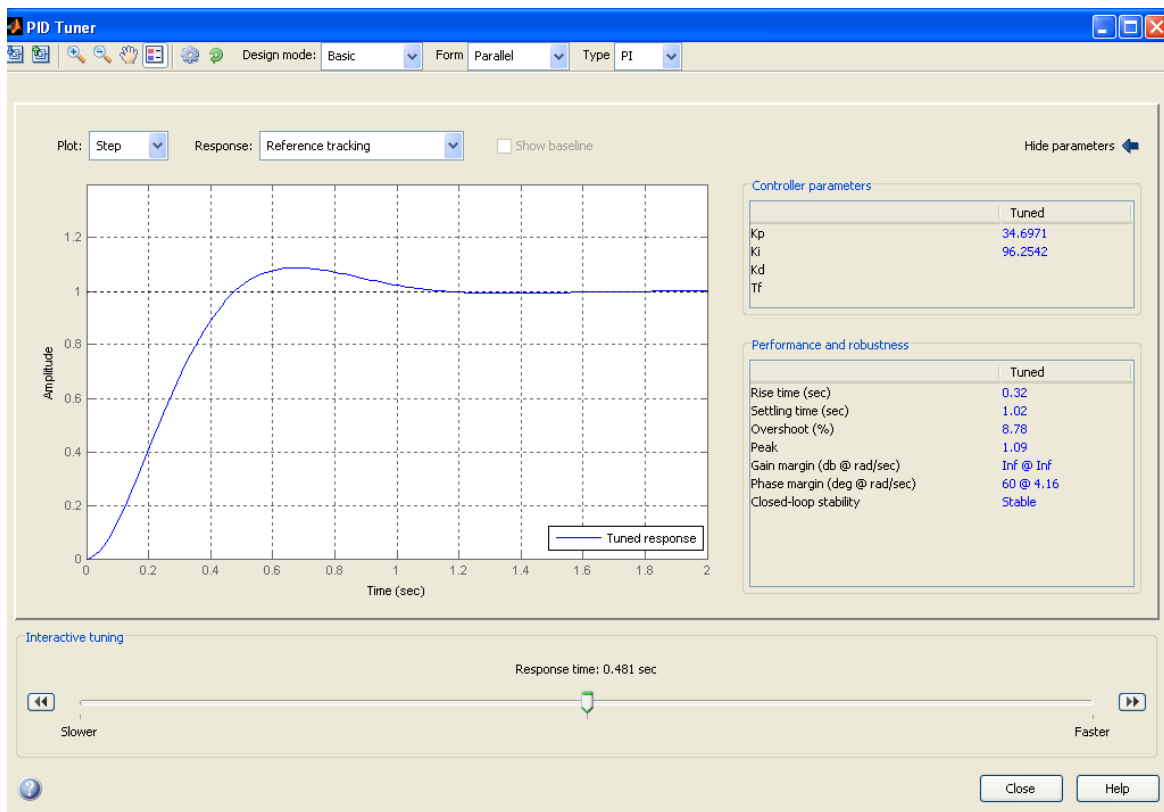


Рис. 7.3.4

7.3.1.5. Введення в САУ пропорційного регулятора

З таблиці 7.3.1 бачимо, що пропорційний регулятор зменшує похибку сталого режиму. Передаточна функція замкнутого кола системи з пропорційним регулятором дорівнює:

$$\frac{X(s)}{F(s)} = \frac{k_p}{s^2 + 10s + (20 + k_p)}$$

Використовуємо функцію `pidtune` для розрахунку коефіцієнтів регуляторів. В програмі MATLAB також є функція `feedback`, яка виконує обчислення передаточної функції замкнутого кола системи за передаточною функцією розімкнутої системи. З її використанням *m*-файл приймає вигляд:

```
>> C=pidtune(W, 'p');
>> Wp=series(W,C);
>> Wpz=feedback(Wp,1,-1);
>> step(Wpz)
```

Виконання цього *m*-файлу в програмі MATLAB дозволяє отримати вихідну реакцію об'єкта керування (рис. 7.3.5).

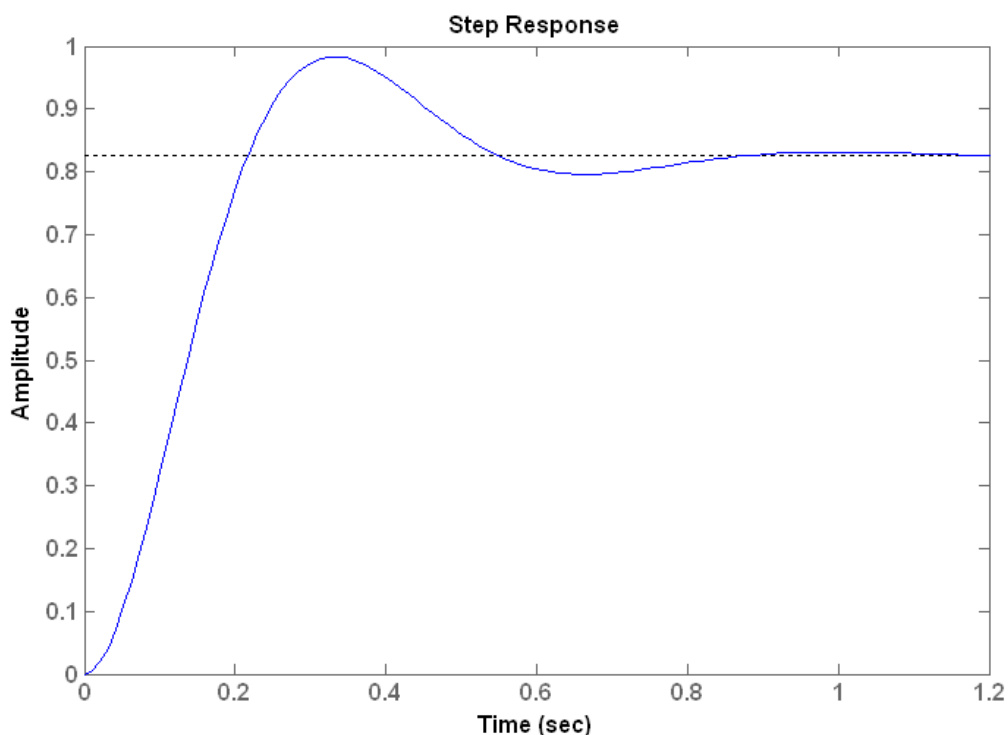


Рис. 7.3.5. Перехідна характеристика САУ з пропорційним регулятором

Пропорційний регулятор зменшує час регулювання до 0,77с і похибку сталого режиму до $\frac{K_p}{1+(K+K_p)} = \frac{94.8517}{1+(94.8517+20)} = 0,82$, але збільшує перерегулювання .

7.3.1.6. Пропорційно-диференційний регулятор в САУ

ПД-регулювання зменшує як перерегулювання, так і час усталення перехідного процесу. Для досліджуємого об'єкта (див. формули (7.3.1) – (7.3.3)) передаточна функція кола з ПД-регулятором записується так:

$$\frac{X(s)}{F(s)} = \frac{k_d s + k_p}{s^2 + (10 + k_d)s + (20 + k_p)}$$

Використовуємо функцію *pidtune* для розрахунку коефіцієнтів регуляторів. Тоді m-файл набуває такого вигляду:

```
>> C=pidtune(W, 'pd');
>> Wpd=series(W,C);
>> Wpdz=feedback(Wpd,1,-1);
>> step(Wpdz)
```

В результаті отримуємо перехідний процес в САУ, зображений на рис. 7.3.6.

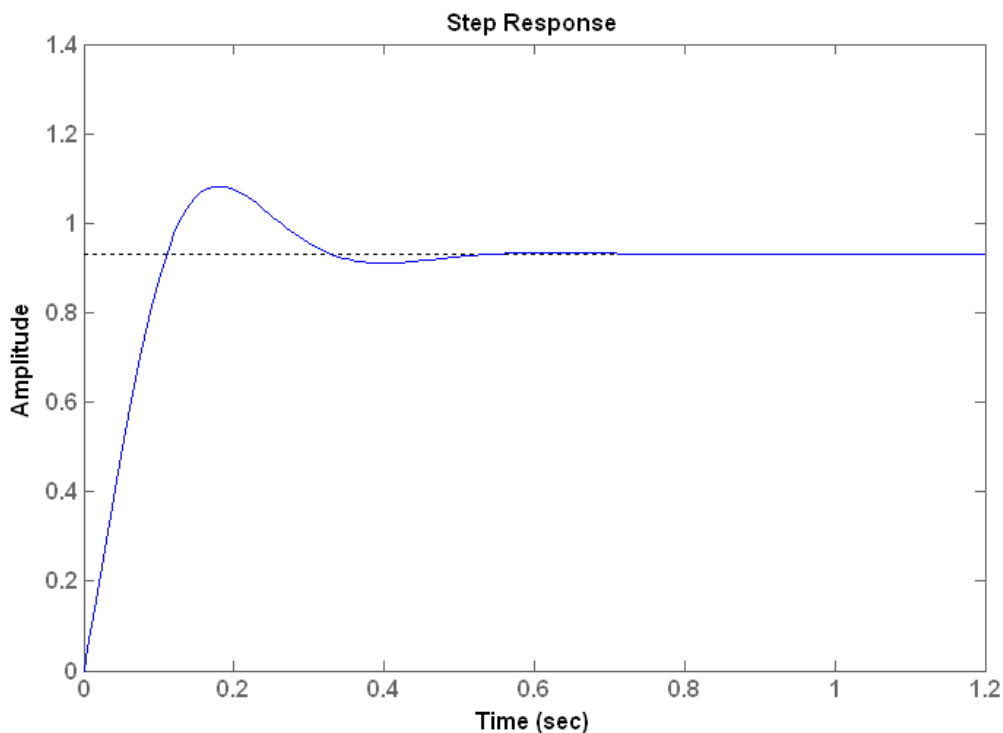


Рис. 7.3.6. Перехідна характеристика САУ з пропорційно-диференційним регулятором

Таким чином, регулювання за похідною зменшує як перерегулювання, так і час усталення перехідного процесу, але має малий вплив на похибку сталого режиму.

7.3.1.7. Пропорційно-інтегральний регулятор в САУ

Інтегральний регулятор зменшує час досягнення максимуму кривої перехідного процесу, збільшує перерегулювання та час усталення перехідного процесу, ліквідує похибку сталого режиму. Для досліджуваного об'єкта (див. формули (7.3.1) – (7.3.3)) передаточна функція замкнутого кола з ПІ-регулятором записується так:

$$\frac{X(s)}{F(s)} = \frac{k_p p + k_i}{s^3 + 10s^2 + (20 + k_p)p + k_i}$$

Використовуємо функцію `pidtune` для розрахунку коефіцієнтів регуляторів. Тоді м-файл набуває такого вигляду:

```
>> C=pidtune(W, 'pi');
>> Wpi=series(W,C);
>> Wpiz=feedback(Wpi,1,-1);
>> step(Wpiz)
```

В результаті отримуємо перехідний процес, зображений на рис. 7.3.7.

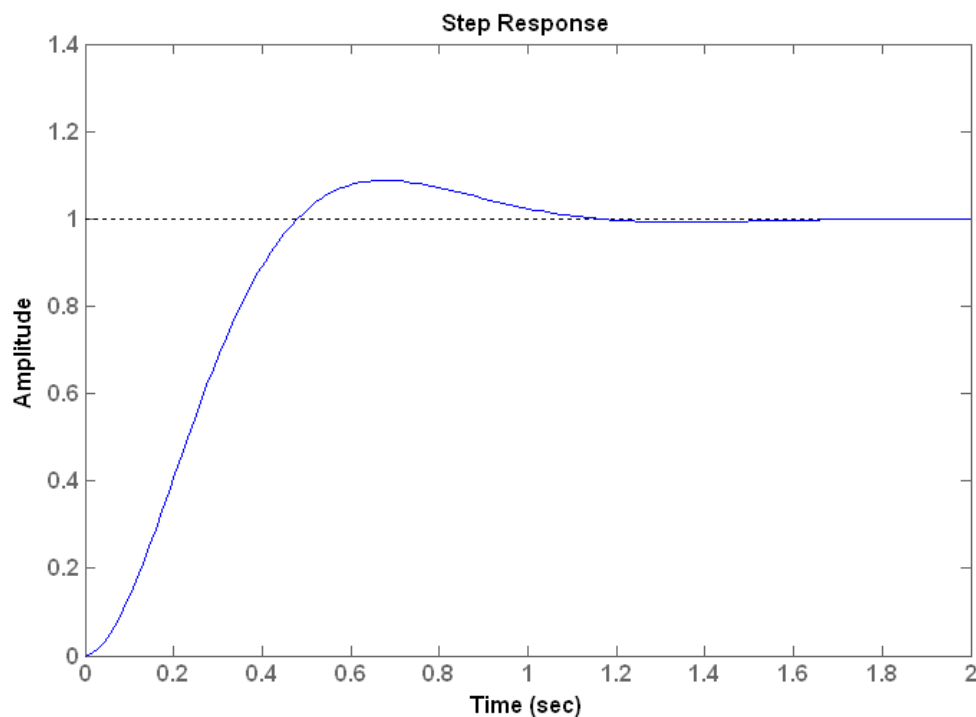


Рис. 7.3.7. Перехідна характеристика САУ з пропорційно-інтегральним регулятором

По графіку перехідного процесу бачимо, що інтегральний регулятор значно зменшив похибку сталого режиму та зменшив перерегулювання, але час встановлення дещо збільшився до 1,02 с в порівняння з попередніми регуляторами.

7.3.1.8. Пропорційно-інтегрально-диференціальний регулятор в САУ

Передаточна функція замкнутого кола системи з ПІД-регулятором має такий вигляд:

$$\frac{X(s)}{F(s)} = \frac{k_d s^2 + k_p s + k_i}{s^3 + (10 + k_d)s^2 + (20 + k_p)s + k_i}.$$

Використовуємо функцію `pidtune` для розрахунку коефіцієнтів регулятора, які забезпечують необхідну якість перехідного процесу. В цьому випадку маємо такий *m*-файл:

```
>> C=pidtune(W, 'pid');
>> Wpid=series(W, C);
>> Wpidz=feedback(Wpid, 1, -1);
>> step(Wpidz)
```

В результаті отримуємо перехідний процес, зображений на рис. 7.3.8.

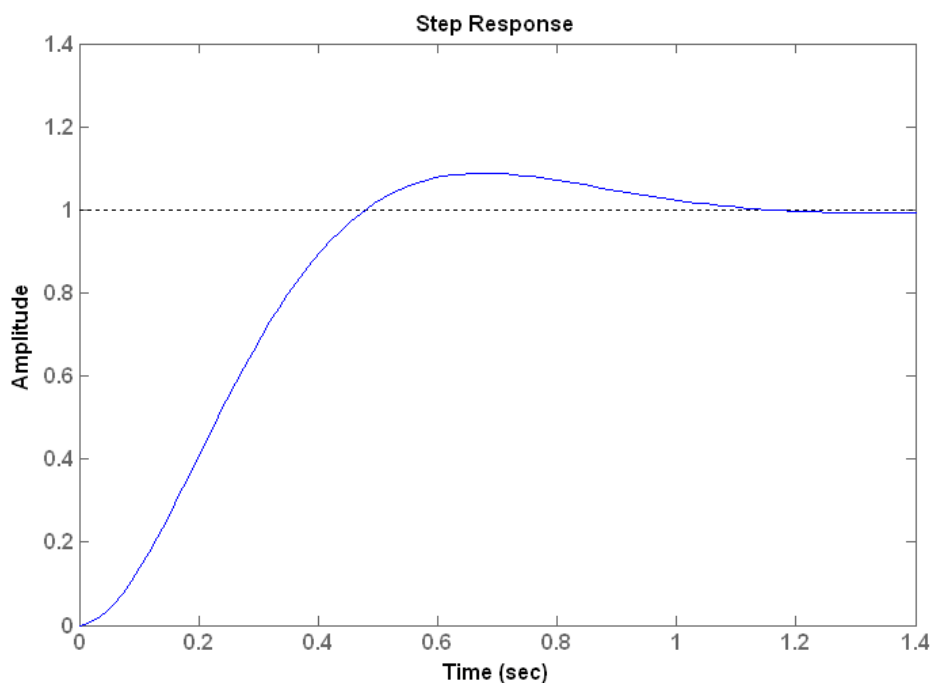


Рис. 7.3.8. Перехідна характеристика САУ з пропорційно-інтегрально-диференційним регулятором

Як, бачимо з графік ПІД-регулятор аналогічно повторює графік ПІ-регулятор. Отже, в даному випадку обираємо ПІ-регулятор.

Таким чином, знайдено параметри системи з ПІД-регулятором, які забезпечують перехідні процеси без перерегулювання, з малим (прийнятним) часом зростання перехідної характеристики та без похибки сталого режиму.

7.3.1.9. Рекомендації по вибору регуляторів

Вибір структури та параметрів регулятора виконується, виходячи з умови отримання необхідних параметрів перехідних процесів в замкнутому

колі синтезуємої системи. Рекомендується наступний порядок дій при визначенні структури та параметрів регулятора:

1. Отримуємо відгук (реакцію) розімкнутого кола системи та приймаємо проектне рішення щодо потрібних удосконалень цієї системи.
2. Встановлюємо пропорційний регулятор для зменшення часу зростання графіка перехідного процесу.
3. Додаємо керування за похідною для зменшення перерегулювання.
4. Додаємо інтегральне керування для ліквідації похибки сталого режиму.
5. Налаштовуємо (підбираємо) кожний із коефіцієнтів k_p , k_i , k_d до отримання необхідних параметрів графіку перехідного процесу, користуючись таблицею 7.3.1.

Не слід одночасно вводити до системи усі три регулятори. Наприклад, якщо П-регулятор забезпечує прийнятний графік перехідного процесу, то непотрібно вводити до системи ще і Д - та І - регулятори. Завжди обмежуються найпростішим регулятором.

7.3.2. Завдання для лабораторної роботи

1. Згідно свого варіанту оберіть структурну схему системи автоматичного управління (додаток 1) та параметри цієї схеми (додаток 2).
2. Введіть передаточну функцію у вигляді полінома за допомогою пакету Simulink та отримайте відгук розімкнутого кола системи та прийміть проектне рішення щодо потрібних удосконалень цієї системи.
3. Введіть передаточну функцію у вигляді полінома в Control System Toolbox та отримайте відгук розімкнутого кола системи, прийміть проектне рішення щодо потрібних удосконалень цієї системи.
4. На основі отриманих графіків визначить час встановлення, перерегулювання, необхідність регулятора, стійкість САУ.
5. Якщо система нестійка, то застосуйте регулятор. Якщо стійка, то покращить перехідну характеристику за допомогою відповідного регулятора.
6. Промодельуйте систему з регулятором за допомогою пакету Simulink та отримайте перехідну характеристику.
7. Промодельуйте систему з регулятором в Control System Toolbox та отримайте перехідну характеристику.
8. Порівняйте отримані графіки перехідної характеристики системи з регулятором та без регулятора.

7.3.3. Зміст звіту

1. Назва та мета роботи.
2. Структурна схема системи автоматичного управління згідно варіанту свого завдання.
3. Результат моделювання даної структурної схеми в Simulink та графік отриманої перехідної характеристики без регулятора.

4. Результат моделювання САУ в Control System Toolbox та графік отриманої перехідної характеристики без регулятора.

5. Результат моделювання даної структурної схеми в Simulink та графік отриманої перехідної характеристики з регулятором.

6. Результат моделювання САУ в Control System Toolbox та графік отриманої перехідної характеристики з регулятором.

7. Розрахунок якості регулювання без регулятора та з регулятором.

8. Аналіз графіків перехідних характеристик системи автоматичного управління, що покращені за допомогою регулятора і отримані різними методами моделювання.

9. Висновки по роботі.

7.3.4. Контрольні питання

1. Як визначити стійкість системи за перехідною характеристикою?

2. Що таке час встановлення та як його визначити?

3. Що собою представляє регулятор?

4. З яких елементів складається регулятор?

5. Який вплив кожного з елементів регулятора на якість перехідних процесів?

6. Що собою являють часові характеристики лінійних систем?

7. Дайте визначення кожної з часових характеристик лінійних систем.

7.4. Лабораторна робота 4. Синтез регуляторів для системи управління частотними методами

Мета роботи: дослідити методи синтезу САУ частотними методами в програмі MATLAB (інструментальні пакети Control System Toolbox та Simulink), визначити показники якості перехідних процесів.

7.4.1. Теоретичні відомості

7.4.1.1. Синтез регуляторів частотними методами

Частотний метод синтезу регуляторів має суттєві переваги перед іншими методами синтезу, особливо при моделюванні передаточних функцій на основі фізичних даних. Частотний метод має дві модифікації: на основі ЛАХ (діаграма Боде по термінології програми MATLAB); на основі АФХ на комплексній площині (діаграма Найквіста за термінологією програми MATLAB). Як відомо, ці два методи відображають одну і ту ж саму інформацію, а різниця між методами полягає в формі представлення цієї інформації.

Відомо, що вихідний сигнал лінійної системи при синусоїдальному вхідному впливі є синусоїда тієї ж частоти з фазовим зсувом, що вносить система. Тому частотний відгук визначається як амплітудні та фазові різниці поміж вхідними та вихідними синусоїдами. При цьому можна використовувати частотний відгук розімкнутої системи для передбачення її поведінки в замкнутому стані (при замиканні зворотного зв'язку).

Для побудови частотних характеристик створюється вектор частот, який має охопити діапазон від нуля до нескінченості, та обчислюється величина передаточної функції на цих частотах. Якщо $W(s)$ – передаточна функція розімкнутого кола системи, а ω – вектор частоти, то будується графік $W(j\omega)$ відносно ω . В зв'язку з тим, що $W(j\omega)$ – функція комплексного числа, то для неї можна побудувати графіки модуля та фази. В результаті отримуємо діаграму Боде. Якщо розташувати цю функцію на комплексній площині, то отримуємо діаграму Найквіста.

Раніше вручну будували асимптотичні ЛАЧХ, підсумовуючи ЛАЧХ окремих ланок. У середовищі MATLAB існують засоби, що дозволяють автоматизувати побудову точних (не асимптотичних) ЛАФЧХ. При цьому можна використовувати накопичений за багато років класичний досвід проектування.

На рис. 7.4.1 представлений графік логарифмічно-амплітудної характеристики САУ та визначено на графіку запаси по амплітуді g_m та фазі φ_m , частоту зрізу $\omega_{зр}$.

Запас стійкості по амплітуді g_m (в дБ) - це відстань від ЛАЧХ до горизонтальної прямої $L_m = 0$ дБ на частоті, на якій фазова характеристика

перетинає пряму $\varphi = -180^\circ$. На цій частоті система повинна мати коефіцієнт посилення менше 1 (або $L_m(\omega) < 0$).

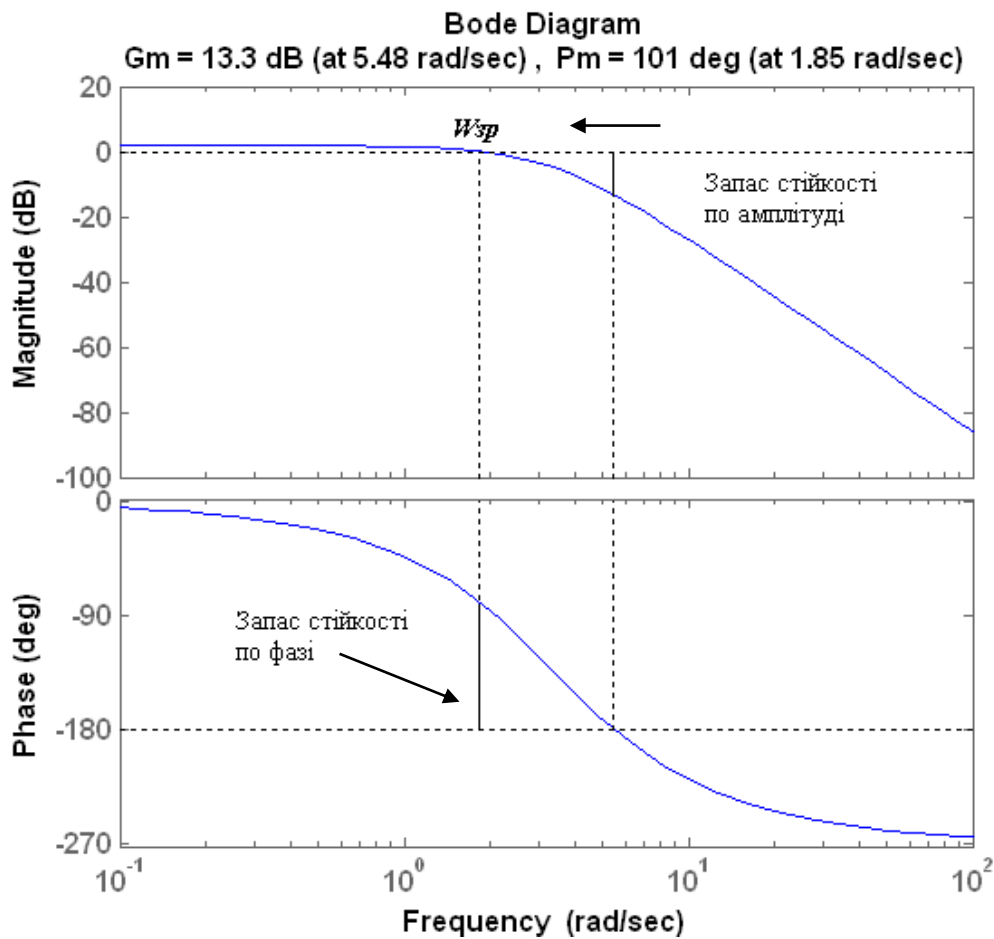


Рис. 7.4.1. Логарифмічна-амплітудна характеристика САУ

Запас стійкості по фазі φ_m (у градусах) - це відстань від частотної характеристики до горизонтальної прямої $\varphi = -180^\circ$ на частоті зрізу ω_{sp} . На цій частоті фазова характеристика повинна мати значення більше -180° .

Допустимим вважається запас по амплітуді не менше 6 дБ і запас по фазі не менше 30 градусів.

Для забезпечення гарної якості перехідних процесів рекомендується, щоб ЛАЧХ перетинала вісь $L = 0$ з нахилом 20 дБ / дек. Це пояснюється тим, що нахил 20 дБ / дек, відповідний аперіодичній ланці, призводить до найменшої коливальності перехідного процесу. Точки переходу (зламу асимптотичної ЛАЧХ) від низькочастотної частини до середньочастотної і далі до високочастотної повинні відстояти від осі $L = 0$ на 12-16 дБ.

Смугою перепускання ω_{bw} називається діапазон частот від 0 до значення частоти, на якій значення ЛАХ замкнутої системи дорівнює -3дБ.

7.4.1.2. Характеристика роботи замкненої системи

Для прогнозування якості роботи замкненої системи на основі відгуків розімкнутої системи використовуються такі положення:

1. Якщо частота зрізу розімкнутої системи менша від частоти, на якій ФЧХ досягає значення -180^0 ($\omega_c < \omega_{pc} = -180^0$), то замкнена система буде стійкою.

2. У замкнутій системі 2-го порядку коефіцієнт демпфування дорівнює запасу по фазі, що віднесений до 100.

3. Для системи 2-го порядку за межу смуги перепускання приблизно можна прийняти частоту власних незгасаючих коливань.

Використаємо вказані властивості при синтезі регулятора для системи, яка відповідає структурній схемі згідно рис. 3.1. В якості регулятора будемо використовувати різні типи регуляторів з передаточною функцією загального виду $G_c(p)$. Нехай об'єкт управління має передаточну функцію

$$W(s) = \frac{1}{s^2 + 10s + 20}.$$

Проектуємий регулятор повинен задовольняти таким вимогам:

- забезпечувати нульову похибку в сталому режимі;
- перерегулювання не повинно перевищувати 40%;
- час усталення кривої перехідного процесу має бути ≤ 2 с.

Задачу синтезу регулятора можна вирішити графічним або чисельним методом. При використанні програми MATLAB графічний метод має явні переваги, тому будемо користуватися графічним методом синтезу регулятора. Спочатку побудуємо ЛАХ, для чого запишемо m-файл:

```
>> W=tf(1,[1 10 20]);  
>> Wz=feedback(W,1,-1)  
>> margin(Wz)
```

Графіки ЛАХ наведено на рис. 7.4.2.

На основі графіків ЛАХ можна визначити деякі характеристики системи. В даному випадку система стійка та має нескінчений запас по амплітуді та по фазі.

Наступним питанням при дослідженні об'єкта управління є похибка сталого режиму. Найпростіше отримати похибку сталого режиму та час усталення шляхом моделювання в програмі MATLAB відгуку замкненої системи на ступеневий вхідний сигнал. Для цього запишемо m-файл:

```
>> W=tf(1,[1 10 20]);  
>> Wzz=feedback(W,1,-1);  
>> step(Wzz)
```

Результатом розрахунків є графік, наведений на рис. 7.4.3.

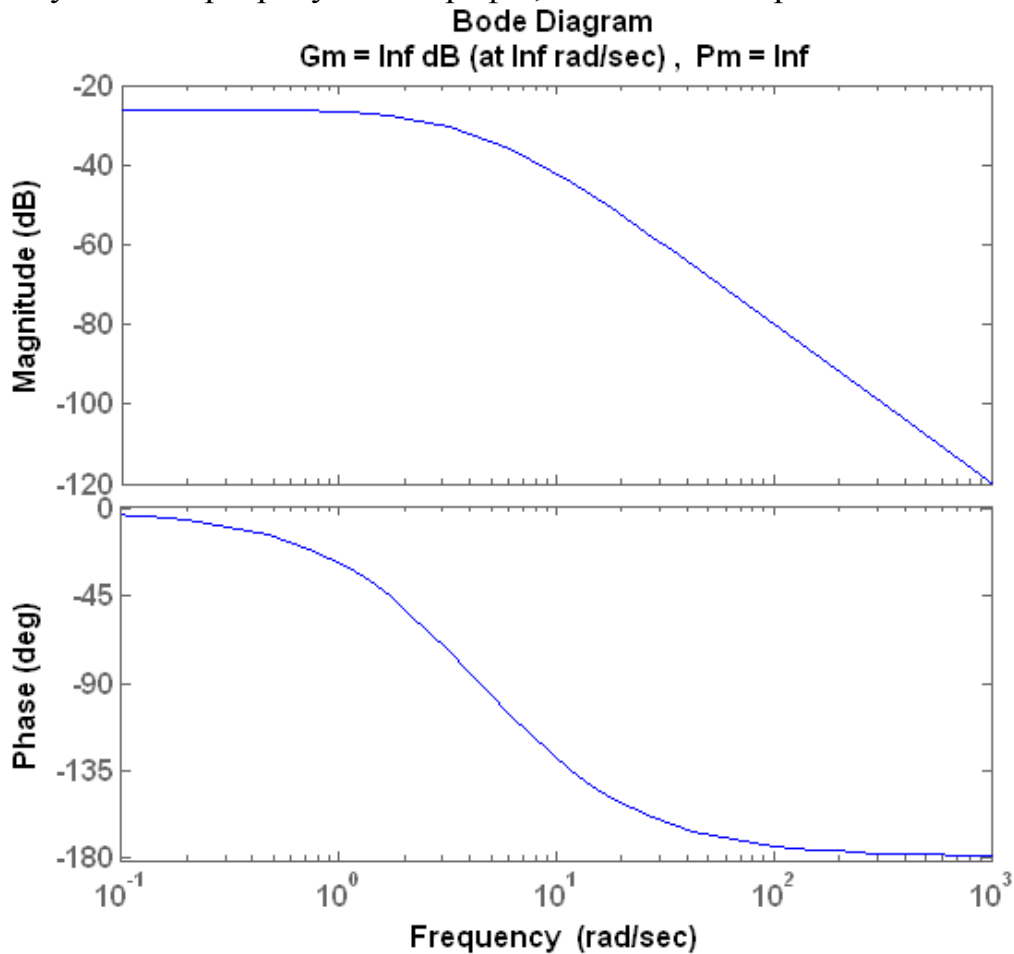


Рис. 7.4.2. Логарифмічна-амплітудна характеристика початкової замкнутої САУ

Час встановлення становить приблизно 1,4 с, а похибка сталого режиму становить 0,95 (див. рис. 7.4.3).

Далі необхідно вибрати регулятор, який буде покращувати вказані характеристики.

На основі результатів на лабораторній роботі №3 вибираємо ПІ-регулятор. Він гарантує нульову статичну похибку при ступеневому вхідному впливі. ПІ-регулятор має нульовий полюс, який можна розташувати на кореневому годографі. Це забезпечує додаткову гнучкість синтезу, яка в свою чергу гарантує виконання технічних вимог до проектуємої системи.

Передаточна функція ПІ-регулятора має вигляд:

$$G(s) = \frac{k(s + a)}{s}$$

За допомогою Control System Toolbox реалізація ПІД-регуляторів проводиться наступним чином:

```
>> pidtool(W)
```

В результаті отримуємо ЛАХ, що зображена на рис. 7.4.4.

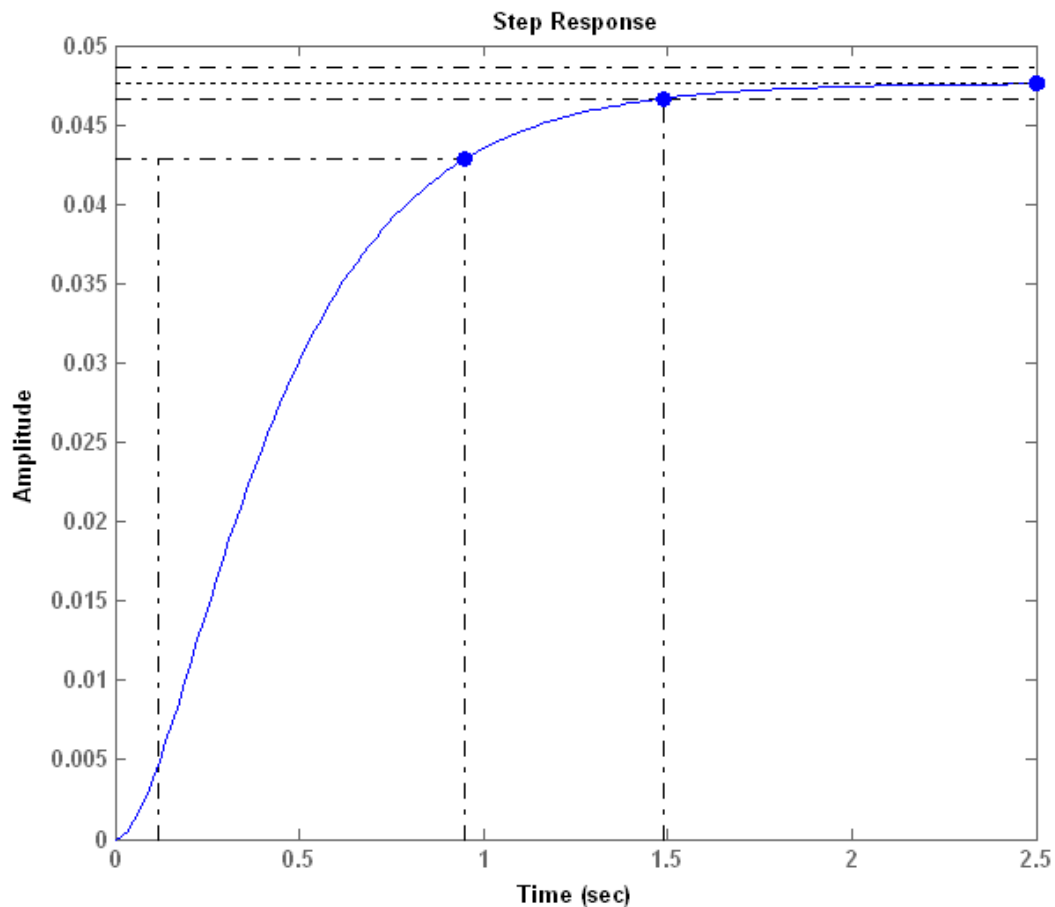


Рис. 7.4.3. Перехідна характеристика початкової замкнутої САУ

Таким чином, ми отримали задовільний результат: система стійка, запас по фазі дорівнює 60° . Час установлення зменшився до 1,02 с, перегулювання дорівнює 8,78% а похибка сталого режиму відсутня. Також, крім функції `pidtool`, є в MATLAB функція `pidtune`, вона розглянута більш детально в лабораторній роботі №3. Перевагою такого методу є те, що вид та коефіцієнти регулятора підбираються автоматично. Недоліком методу є те, що точність підбору коефіцієнтів залежить від точності розрахунку в MATLAB функції `pidtune`. Також потрібно окремими командами будувати імпульсні та передаточні функції.

Таким чином, спрощений варіант синтезу регулятора полягає в підборі структури та параметрів регулятора, які повинні забезпечити визначені в технічному завданні на проектування показники динамічних властивостей проектуємої системи.

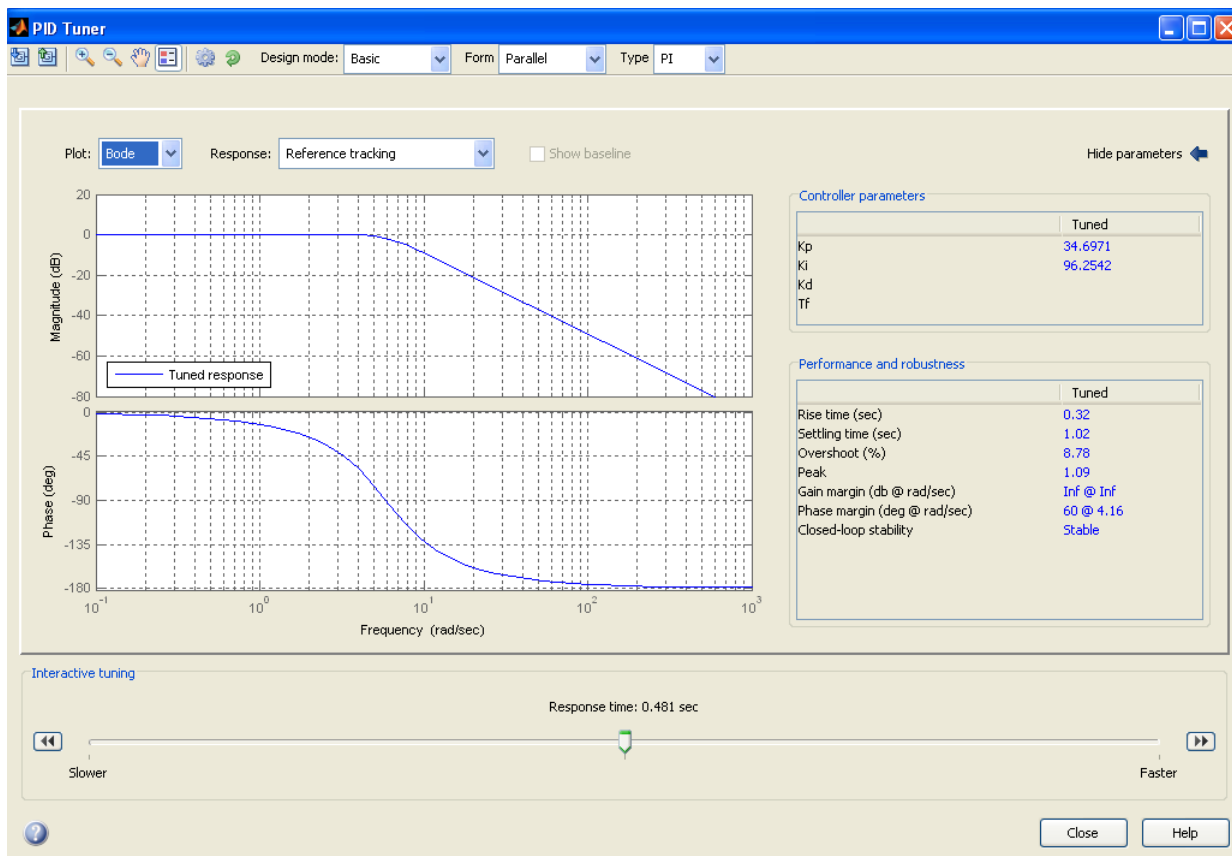


Рис. 7.4.4. Результат автоматичного моделювання САУ з регулятором частотним методом

7.4.2. Завдання для лабораторної роботи

1. Згідно свого варіанту обрати структурну схему системи автоматичного управління (додаток А).
2. Визначити параметри цієї схеми (додаток Б).
3. Ввести передаточну функцію у вигляді полінома в пакет Control System Toolbox та отримати ЛАХ систем управління.
4. На основі отриманої характеристики визначити смугу пропускання, запас по фазі, перерегулювання, стійкість системи управління.
5. Якщо система нестійка, то необхідно застосувати регулятор. Якщо стійка, то необхідно покращити ЛАХ (задачу синтезу регулятора вирішити графічним методом).
6. Для оцінки отриманих параметрів регулятора виконати моделювання системи і отримати перехідну характеристику замкненої системи.
7. Порівняти отримані графіки ЛАХ системи з регулятором та без регулятора.

7.4.3. Зміст звіту

1. Назва та мета роботи.
2. Структурна схема системи автоматичного управління згідно свого варіанту завдання.

3. Результати моделювання в пакеті Control System Toolbox та графік ЛАХ системи без регулятора.

4. Результат моделювання в пакеті Control System Toolbox та графік ЛАХ системи з регулятором.

6. Розрахунок смуги пропускання, запасу по фазі, перерегулювання, визначення стійкості систем управління.

7. Аналіз графіків ЛАХ системи автоматичного управління, покращених за допомогою регулятора.

8. Висновки по роботі.

7.4.4. Контрольні питання

1. Як визначити на графіку ЛАХ запас стійкості по амплітуді?

2. Як визначити на графіку ЛАХ запас стійкості по фазі?

3. Що таке смуга пропускання?

4. Що собою представляє регулятор?

5. З яких елементів складається регулятор?

6. Який вплив кожного з елементів регулятора на якість перехідних процесів?

7. Що собою являють частотні характеристики лінійних систем?

8. Дайте визначення кожної з частотних характеристик лінійних систем.

7.5. Лабораторна робота 5. Методи перетворення неперервної системи автоматичного управління в цифрову систему

Мета роботи: ознайомитися з можливостями програми MATLAB по моделюванню цифрових систем управління та дослідити їх характеристики

7.5.1. Теоретичні відомості

7.5.1.1. Дискретні системи

Дискретні системи - це системи, що містять елементи, які перетворюють безперервний сигнал в дискретний. У дискретних системах сигнали описуються дискретними функціями часу.

Класифікація дискретних систем

Головним напрямком розвитку систем автоматизації в останні десятиріччя є широке використання засобів обчислювальної техніки та мікропроцесорних пристроїв, об'єднаних в мережі різного рівня і призначення. За характером сигналів такі системи є дискретними, тобто ці сигнали є послідовністю імпульсів, які несуть в собі всю необхідну інформацію. Дискретні системи мають ряд переваг перед неперервними (аналоговими):

- можливість багатоточкового керування з багатократним використанням ліній зв'язку, по яких одночасно передається множина сигналів за рахунок їх особливостей: імпульс – пауза тощо;
- підвищена завадостійкість за рахунок того, що завада діє лише на протязі імпульсу, який може бути як завгодно коротким. В паузах між імпульсами система розімкнена і перешкода на неї не діє.

В дискретних системах об'єкт керування, як правило, неперервний за своєю природою, тому відбувається перетворення неперервного сигналу в дискретний, тобто його квантування за рівнем та за часом. Квантування – процес перетворення безперервного сигналу в дискретний.

Види квантування сигналів, що застосовуються, лежать в основі класифікації дискретних систем.

В релейних (позиційних) системах відбувається квантування за рівнем (рис. 7.5.1,а), коли виділяється значення $\Delta X = const$, і для цих значень визначається рівень неперервного сигналу.

В імпульсних системах здійснюється квантування за часом при $\Delta t = const$ (рис. 7.5.1,б). Для збереження певного рівня сигналу між сусідніми точками використовуються екстраполятори: нульового порядку (зберігають сигнал постійним); першого та другого порядків (змінюють сигнал за лінійним чи нелінійним законами).

В цифрових системах здійснюється змішане квантування (рис. 7.5.1,в) – за часом та за рівнем $(\Delta t, \Delta X)$. Значення квантового сигналу береться на перетині відповідних ліній.

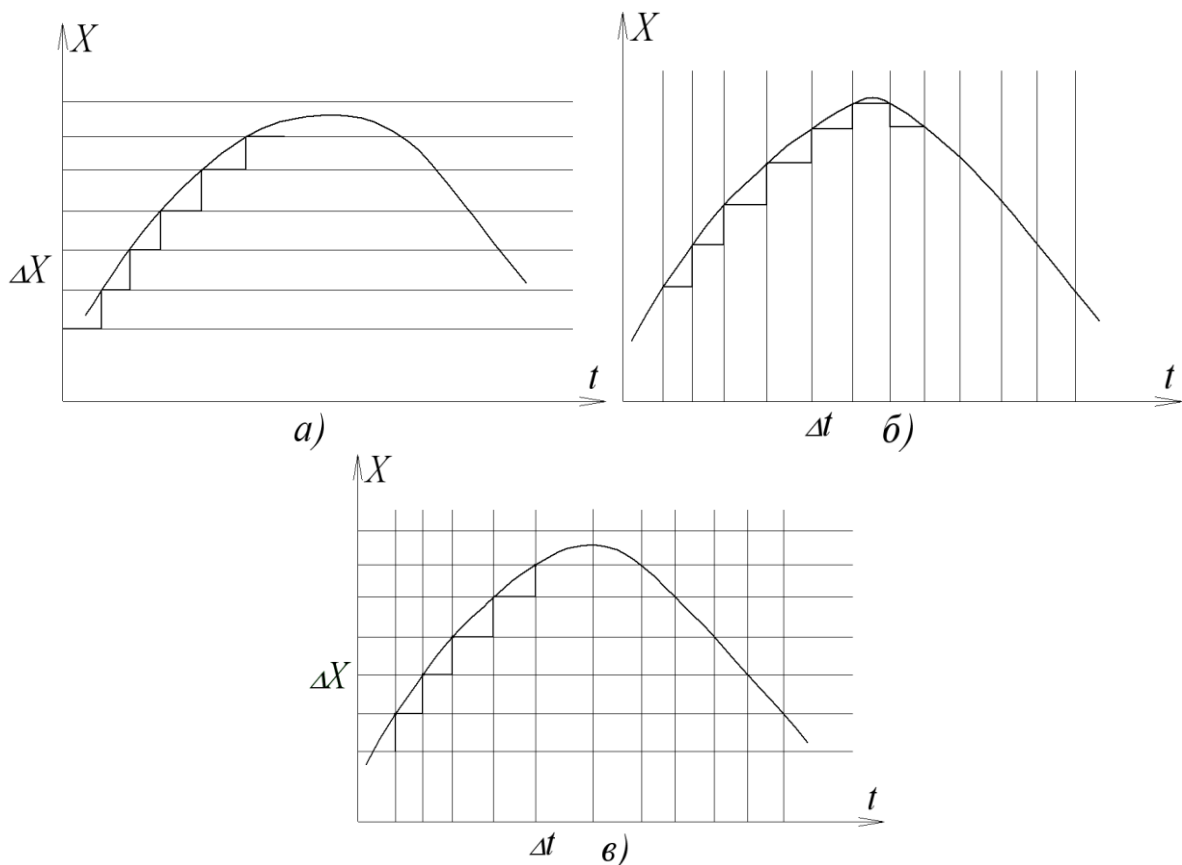


Рис. 7.5.1. Види квантування сигналу: а – за рівнем, б – за часом, в – за часом та рівнем.

Загальна характеристика імпульсних систем (ІС)

Процес квантування неперервного сигналу за часом – це є імпульсна модуляція, тобто перетворення неперервного вхідного сигналу в послідовність, наприклад, амплітудно-модульованих імпульсів з обвідною, яка співпадає з вхідним сигналом.

Вихідний сигнал імпульсного елемента (рис. 7.5.2) характеризується кількома основними параметрами: A – амплітуда; τ_{iM} - тривалість (ширина) імпульсу; T_n – період повторення імпульсів; $T_n - \tau_{iM}$ - пауза; $\gamma = \frac{\tau_{iM}}{T_n}$ - шпарність імпульсу.

Імпульсна модуляція – змінювання одного з параметрів вихідних імпульсів (модулюємого) у функції величини вхідного сигналу (модулюючого).

Може змінюватись (модулюватись) амплітуда, ширина імпульсу, пауза. Відповідно виділяють види імпульсної модуляції:

- амплітудно-імпульсна (АІМ) – рис. 7.5.3, а;
- широтно-імпульсна (ШІМ) – рис. 7.5.3, б;
- часо-імпульсна (ЧІМ) - рис. 7.5.3, в.

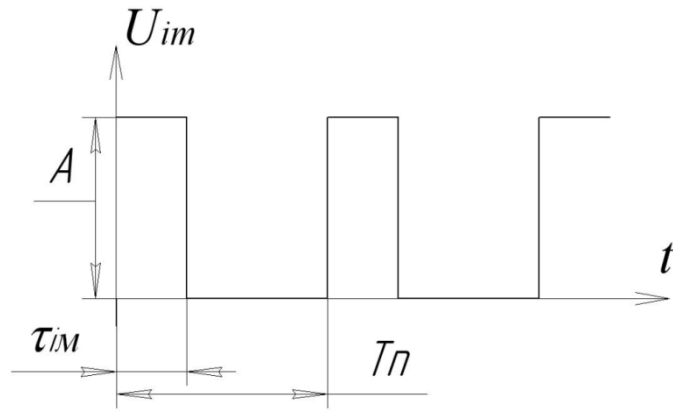


Рис. 7.5.2. Вихідний сигнал імпульсного елемента

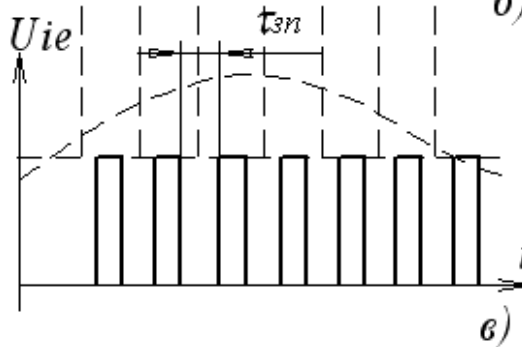
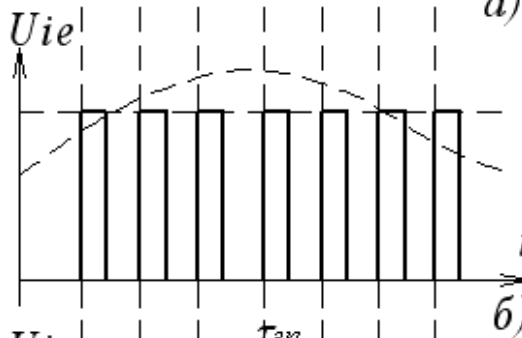
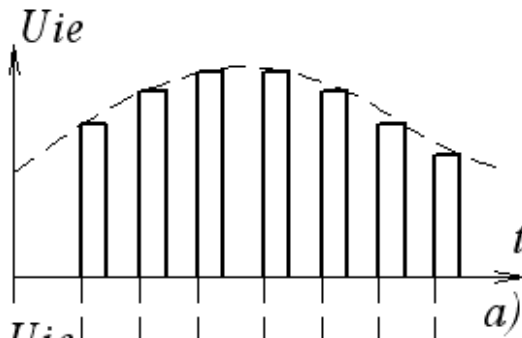
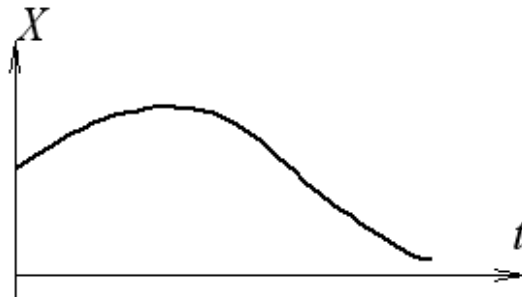


Рис. 7.5.3. Види імпульсної модуляції: а - амплітудно-імпульсна, б - широтно-імпульсна, в - часо-імпульсна.

На рис. 7.5.3 через x позначено вхідний сигнал імпульсного елемента (ІЕ).

При АІМ змінюється амплітуда $A=f(x)$, $\tau_{im}, T_n - const$.

При ШІМ: $\tau_{im}=f(x)$, $A, T_n - const$. При ЧІМ (різновид – фазоімпульсна модуляція): $\tau_{zn}=f(x)$, $A, \tau_{im}, T_n - const$.

Математичний опис цифрових та імпульсних систем з АІМ

Для математичного опису цифрових та ІС, всі сигнали, в тому числі в неперервній частині, розглядаються в дискретні моменти часу $t = 0T_n; 1T_n; 2T_n; \dots; iT_n; \dots; \infty$. Неперервні сигнали подаються у вигляді решітчастих функцій (рис. 7.5.4):

$$x^*(t) = x(iT_n) = x(t) \Big|_{t=iT_n}. \quad (7.5.1)$$

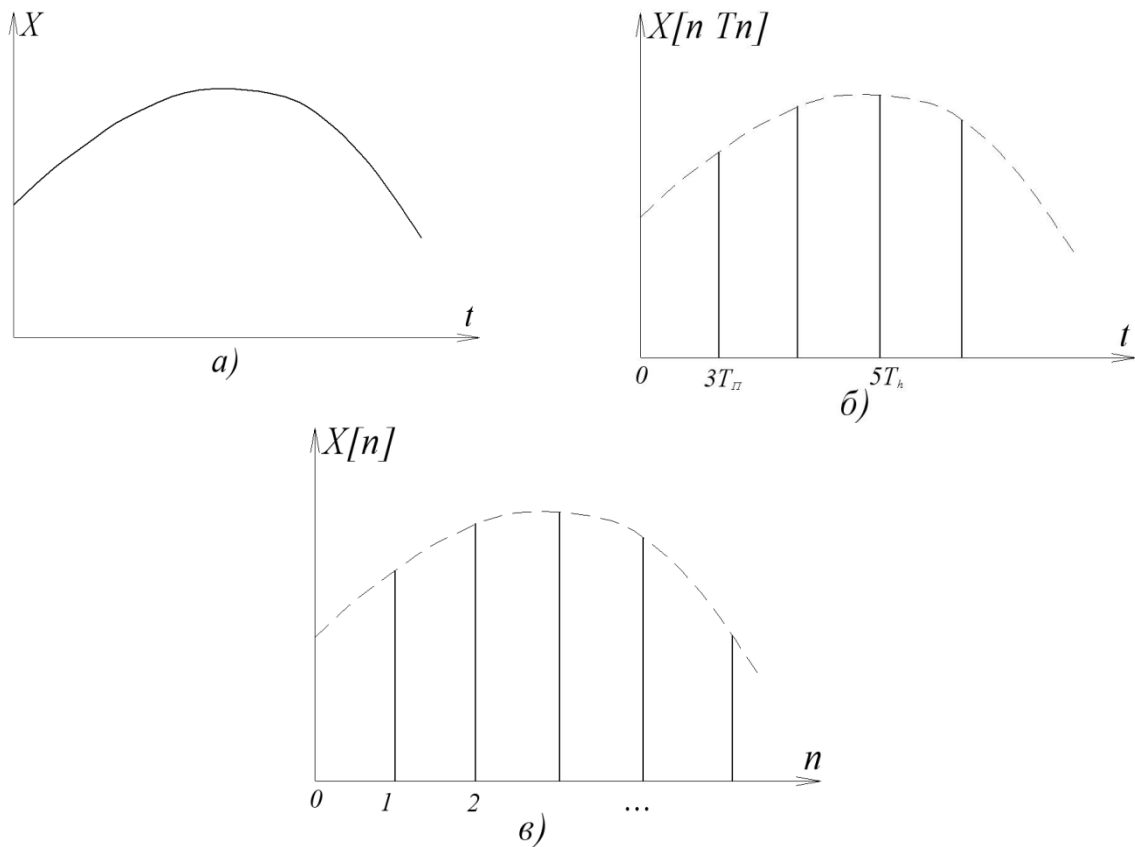


Рис. 7.5.4. Решітчасті функції: а – неперервний сигнал; б, в – форми представлення решітчастих функцій.

Між дискретними значеннями аргументу решітчаста функція дорівнює нулю, а неперервний сигнал є обвідною для решітчастої функції. Послідовність неодиначних імпульсів, які утворюють решітчасту функцію на інтервалі $0 \leq iT_n \leq \infty$, можна подати у вигляді нескінченного ряду:

$$x^*(t) = \sum_{i=0}^{\infty} x(iT_n) \cdot \delta(t - iT_n), \quad (7.5.2)$$

де: $\delta(t - iT_n)$ - зміщена δ - функція, яка існує лише в моменти часу $t=iT_n$ і дорівнює нулю при всіх інших значеннях t . Для решітчастої функції існує дискретне перетворення Лапласа:

$$x^*(p) = L(x^*(t)) = \sum_{i=0}^{\infty} x(iT_n) \cdot e^{-piT_n}. \quad (7.5.3)$$

Вираз (7.5.3) отримано з урахуванням того, що зображення суми оригіналів дорівнює сумі їх зображень, а зображення зміщеної δ -функції дорівнює e^{-piT_n} . Дискретне перетворення Лапласа включає трансцендентний множник e^{-piT_n} , тому зображення $X^*(p)$ та передаточні функції стають ірраціональними функціями аргументу p , що утруднює їх використання. Для отримання передаточних функцій в дрібно-раціональній формі (як для неперервних систем) замінюють аргумент

$$z = e^{pT_n} \quad (7.5.4)$$

і отримують зручне для використання z – перетворення решітчастої функції:

$$X(z) = L(x(iT_n)) = \sum_{i=0}^{\infty} x(iT_n) \cdot z^{-i}. \quad (7.5.5)$$

В табл. 7.5.1 наведені z – зображення для деяких функцій часу.

Таблиця 7.5.1

z – зображення функцій часу

№ п/п	$X(t)$ ($t \geq 0$)	$X(iT_n)$	$X(p)$	$X(z)$
1.	$\delta(t)$	$\delta(iT_n)$	1	1
2.	$1(t)$	$1(iT_n) = \sum_{i=0}^{\infty} \delta(t - iT_n)$	$\frac{1}{p}$	$\frac{z}{z-1}$
3.	t	(iT_n)	$\frac{1}{p^2}$	$\frac{T_n \cdot z}{(z-1)^2}$
4.	t^2	$(iT_n)^2$	$\frac{2!}{p^3}$	$\frac{T_n^2 z(z+1)}{(z-1)^3}$
5.	$e^{-\alpha t}$	$e^{-\alpha \cdot iT_n} = d^i$ ($d = e^{-\alpha T_n}$)	$\frac{1}{p + \alpha}$	$\frac{z}{z-d}$

Зручність z – перетворення полягає в тому, що сама форма запису дає простий спосіб прямого та зворотного перетворення:

- для знаходження z – перетворення за відомою функцією часу необхідно кожне дискретне значення $X(iT_n)$ помножити на z^{-i} , а потім згорнути отриманий степеневий ряд в скінчену суму;
- для знаходження оригіналу за відомим зображенням $X(z)$ необхідно зображення подати у вигляді степеневого ряду за спадаючими степенями z^{-i} , а отримані при цьому числові коефіцієнти ряду i є дискретними значеннями $X(iT_n)$ сигналу $X(t)$.

z – перетворення має властивості, аналогічні властивостям звичайного перетворення Лапласа:

- лінійність:

$$L(a_1x_1(t) \pm a_2x_2(t)) = a_1x_1(z) \pm a_2x_2(z), \quad (7.5.6)$$

- теорема про початкове значення оригіналу:

$$\lim_{i \rightarrow 0} X(iT_n) = \lim_{z \rightarrow \infty} X(z), \quad (7.5.7)$$

- теорема про скінчене значення оригіналу:

$$\lim_{i \rightarrow \infty} X(iT_n) = \lim_{z \rightarrow 0} \frac{z-1}{z} X(z), \quad (7.5.8)$$

- теорема про зміщення аргументу оригінала (теорема запізнювання):

$$L(X(iT_n - lT_n)) = X(z) \cdot z^{-l}. \quad (7.5.9)$$

Стійкість та якість цифрових та імпульсних систем

За динамічними властивостями цифрові та імпульсні системи з АІМ багато в чому аналогічні неперервним системам, що дає можливість застосовувати аналоги методів дослідження неперервних систем. Імпульсна або цифрова система буде стійкою, коли вільна складова перехідного процесу $X_B(iT_n)$ з часом затухає, тобто:

$$\lim_{i \rightarrow \infty} X_B(iT_n) = 0. \quad (7.5.10)$$

Для дослідження стійкості імпульсних та цифрових систем можна використовувати алгебраїчні та частотні критерії стійкості, застосувавши відповідні z -перетворення.

Імпульсний елемент не впливає на стійкість розімкненого контуру, але для замкненої системи необхідно врахувати таке:

- при малих періодах повторення T_n частотна характеристика розімкненого контуру співпадає з частотною характеристикою неперервної частини, яка визначає стійкість імпульсної системи;
- при збільшенні періоду повторення в більшості систем зменшується граничний передаточний коефіцієнт, погіршуються динамічні властивості;
- в окремих випадках (структурно-нестійкі неперервні системи, системи із запізнюванням) імпульсний елемент справляє стабілізуючу дію. Для таких систем рекомендується обирати період повторення T_n з умови:

$$T_n \geq \frac{\pi}{\omega_0}, \quad (7.5.11)$$

де: ω_0 - частота, при якій АФХ неперервної частини перетинає додатну уявну вісь.

Для оцінки **якості** імпульсних та цифрових систем використовуються такі ж показники, як і для неперервних: точність в усталених режимах, тривалість перехідного процесу та інше.

Тривалість і перерегулювання оцінюють безпосередньо за перехідною характеристикою. Для її отримання записують z -зображення вихідної величини при одиничному ступінчастому сигналі:

$$X(z) = \frac{z}{z-1} \cdot W(z). \quad (7.5.12)$$

і за зображенням знаходять оригінал – решітчасту функцію $x(iT_n)$. В простих випадках для цього достатньо таблиць оберненого z – перетворення, розклавши попередньо зображення $X(z)$ на прості дроби.

В більш складних випадках розкладають функцію $X(z)$ в степеневий ряд за від’ємними степенями z (діленням чисельника на знаменник):

$$X(z) = \sum_{i=0}^{\infty} C_i z^{-i} = C_0 + C_1 z^{-1} + C_2 z^{-2} + \dots + C_l z^{-l} + \dots \quad (7.5.13)$$

Точність імпульсної системи оцінюють за усталеним значенням сигналу похибки:

$$\Delta x_{уст}(\infty) = \lim_{z \rightarrow 1} \frac{z-1}{z} W_{зд}^n(z) \cdot x_{зд}(z). \quad (7.5.14)$$

При ступінчастому сигналі $x_{зд}(t) = aI(t)$ усталена похибка буде:

$$\Delta x_{уст}(\infty) = \lim_{z \rightarrow 1} \frac{z-1}{z} \cdot \frac{1}{1+W_{роз}(z)} \cdot \frac{az}{z-1} = \frac{a}{1+W_{роз}(1)}. \quad (7.5.15)$$

Видно, що при ступінчастому сигналі похибка дорівнює нулю, якщо передаточна функція $W_{роз}(z)$ має хоча б один полюс, який дорівнює одиниці. При лінійному сигналі для цього потрібно не менше 2-х полюсів.

Цифрові системи

В цифрових системах відбувається квантування сигналів за часом і рівнем. Квантування за часом робить цифрову систему дискретною, а квантування за рівнем – нелінійною. В цифрових системах є пристрої, які перетворюють неперервні сигнали в цифрові коди і виконують математичні операції над цими кодами. Цифровий регулятор виконує властиві йому операції і видає результати у дискретні моменти часу $t = T_n, 2T_n, 3T_n \dots$. В інтервалах між цими моментами на виході регулятора зберігається певний сигнал, тобто вихідний сигнал – ступінчаста функція

$x(iT_n)$, яка відповідає квантуванню за часом. Квантування за рівнем обумовлюється тим, що внаслідок цифрової подачі інформації вихідний сигнал може набувати лише певних фіксованих рівнів, які відрізняються один від одного на величину q . Ця величина відповідає одиниці молодшого розряду цифрового регулятора, тобто неперервний сигнал подається у виді:

$$x(t) = x^*(iT_n) + \sigma, \quad (7.5.16)$$

де: $|\sigma| < q$, а $x^*(iT_n)$ містить ціле число рівнів q . При малих q впливом квантування за рівнем на динаміку систем можна знехтувати, тобто вважати $q \rightarrow 0$. У загальному випадку для дослідження цифрових систем можна застосувати математичний апарат, який використовується для лінійних імпульсних систем з амплітудно-імпульсною модуляцією: z – перетворення і різницеві рівняння.

В системах автоматичного керування технологічними об'єктами функції регулятора виконує мікропроцесорний контролер. Така система відноситься до неперервно-дискретних і описується диференційними і різницевиими рівняннями, а також включає функціональні залежності, які відображають перетворення сигналів з неперервної форми в дискретну і навпаки. Така структура математичного опису громізка і незручна.

Більш зручним методом є заміна диференційних рівнянь різницевиими, тоді в цілому аналіз і синтез систем виконується методами теорії неперервних систем, а синтезований регулятор реалізується в цифровому виді. При цьому необхідно врахувати, що при вказаних замінах виникають похибки, які можуть привести до різних оцінок, наприклад, щодо стійкості.

Узагальнена функціональна структура цифрової системи показана на рис. 7.5.5.

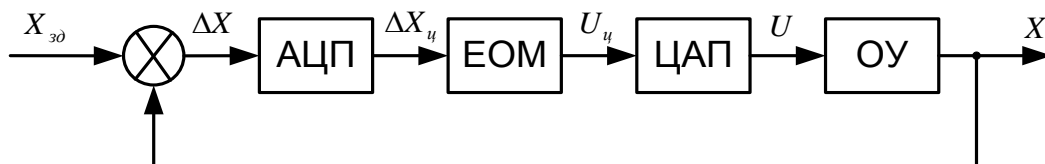


Рис. 7.5.5 Функціональна структура цифрової системи управління

Об'єкт управління ОУ – неперервна частина системи (НЧ). Аналого-цифровий перетворювач АЦП призначений для отримання з неперервного сигналу цифрового коду, який обробляється в ЕОМ або мікропроцесорному контролері (МПК). Для формування сигналу керування U , який поступає на об'єкт, необхідно забезпечити зворотнє перетворення, для чого призначений цифро – аналоговий перетворювач ЦАП. Перетворювач АЦП включає імпульсний елемент для квантування за часом і квантувач за рівнем. В результаті отримують число у вигляді коду, як правило, двійкового, яке подається в ЕОМ (МПК). Після перетворення за певними алгоритмами результат видається у вигляді чисел $U_{ц}(iT_n)$. Перетворювач ЦАП складається з квантувача за рівнем, ідеального δ -імпульсного елемента і формуючого

елемента (екстраполятора). Крім екстраполяторів нульового порядку, які утримують сигнал на постійному рівні між сусідніми імпульсами, застосовуються також екстраполятори першого порядку, сигнал яких змінюється за лінійним законом, і другого – за квадратичною параболою.

Число рівнів квантування вхідного і вихідного сигналів різне: на вході необхідно забезпечити високу точність обробки сигналів, і число рівнів вхідного сигналу визначається розрядністю ЕОМ (МПК). Вихідний сигнал може мати мінімальну кількість рівнів, тобто сигнал може бути релейним.

В порівнянні з аналоговими (неперервними) системами цифрові системи керування мають ряд особливостей, які визначають їх динаміку:

- квантування сигналів за часом і рівнем;
- цифро-аналогове і аналого-цифрове перетворення. В сучасних системах можна забезпечити необхідну точність цих перетворень, але необхідно врахувати, що в алгоритмах керування використовуються прирости вхідних та вихідних сигналів. Це потребує узгодження розрядності технічних засобів, швидкодії, періоду опитування датчиків тощо;
- часовий зсув між вхідним сигналом і видачею сигналів керування, тобто наявність запізнювання. Це має особливе значення, коли здійснюється багатооб'єктне керування, виконується множина необхідних алгоритмів.

7.5.1.2. Методи переходу від безперервної системи до цифрової системи

Перехід від безперервної системи до цифрової може відбуватися декількома методами:

- c2d перетворення (імпульсний елемент з екстраполятором);
- білінійне перетворення;
- impinvar перетворення на основі інваріантної імпульсної характеристики.

За допомогою програми MATLAB вказані види перетворення відбувається наступним чином (рис. 7.5.6-7.5.14):

```
>> W=tf([1],[1 10 20]);
>> Wd = c2d(W,0.01,'zoh')
Transfer function:
4.837e-005 z + 4.678e-005
-----
z^2 - 1.903 z + 0.9048
Sampling time: 0.01>> impulse(Wd,10)
>>step(Wd)
>>impz(Wd)
>>bode(Wd)
```

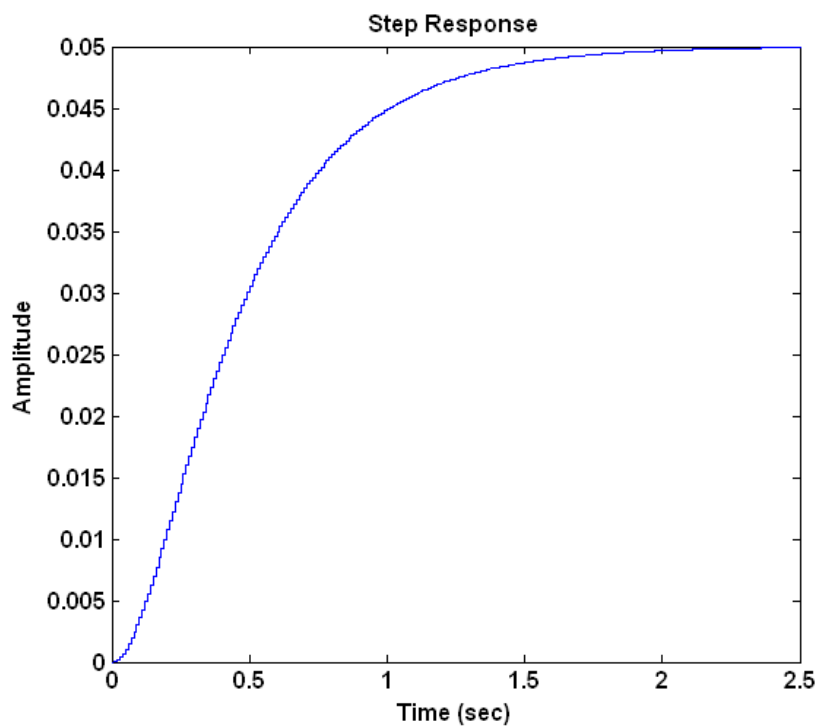



Рис. 7.5.6. Перехідна характеристика цифрової САУ на основі c2d перетворення

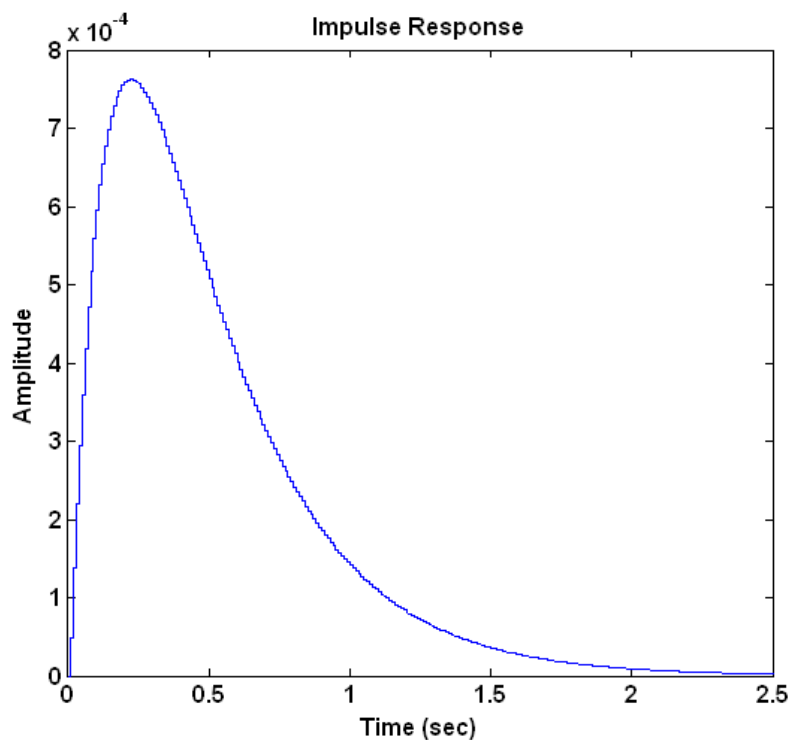


Рис. 7.5.7. Імпульсна характеристика цифрової САУ на основі c2d перетворення

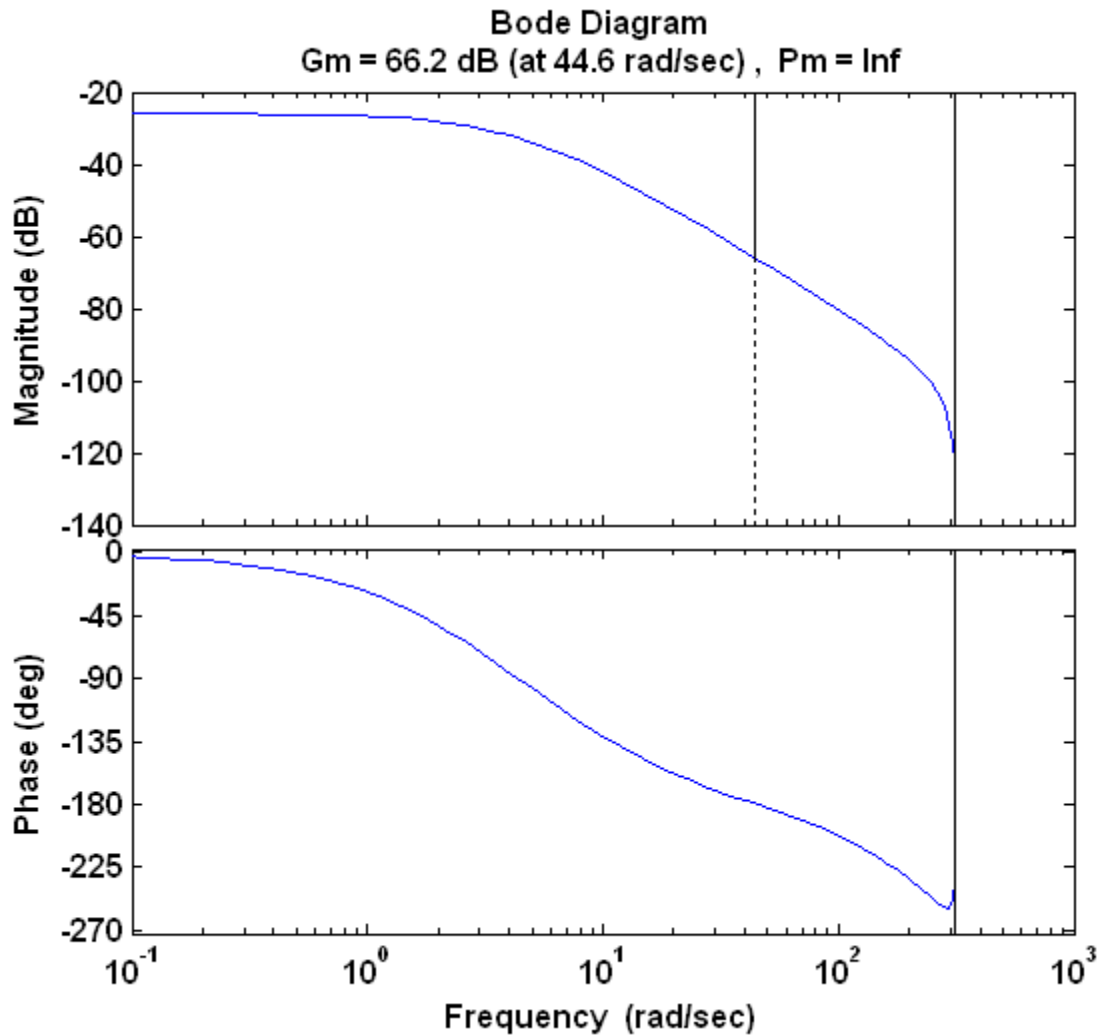


Рис. 7.5.8. Логарифмічна амплітудна характеристика цифрової САУ на основі c2d перетворення

```
>> num=1; den=[1 10 20];
>> [numd1,dend1] = bilinear(num,den,200);

>> Wd1=tf(numd1, dend1, 0.01)

Transfer function:
6.097e-006 z^2 + 1.219e-005 z + 6.097e-006
-----
z^2 - 1.951 z + 0.9512

Sampling time: 0.01
>> impulse(Wd1,10)
>> step(Wd1,10)
>> bode(Wd1)
```

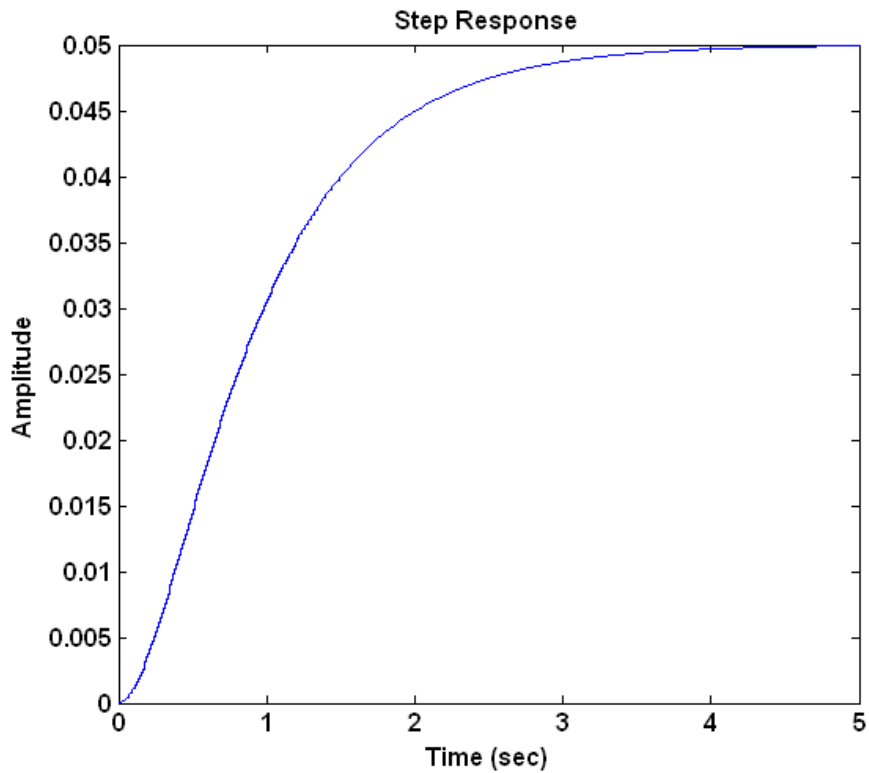


Рис. 7.5.9. Перехідна характеристика цифрової САУ на основі білінійного перетворення

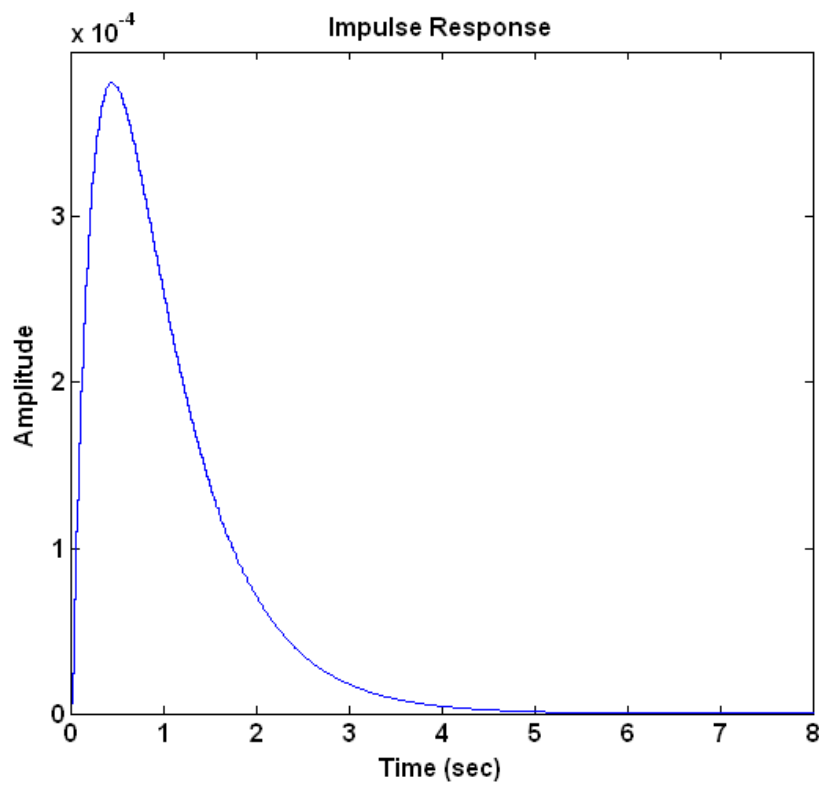


Рис. 7.5.10. Імпульсна характеристика цифрової САУ на основі білінійного перетворення

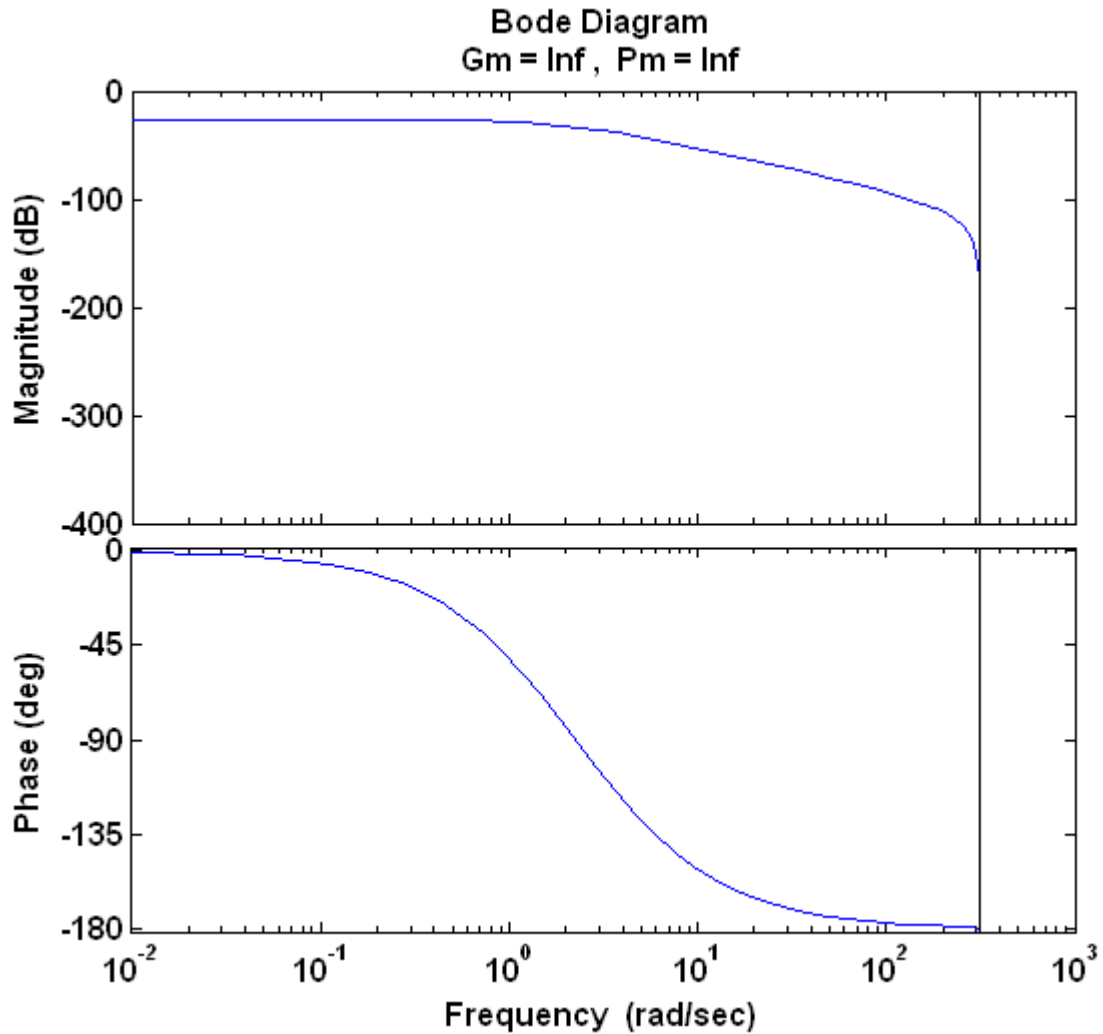


Рис. 7.5.11. Логарифмічна амплітудна характеристика цифрової САУ на основі білінійного перетворення

```
>> [numd2, dend2]=impinvar (num, den, 200);
```

```
>> Wd2 =tf(numd2, dend2, 0.01)
```

```
Transfer function:
```

```
2.438e-005
```

```
-----  
z^2 - 1.951 z + 0.9512
```

```
Sampling time: 0.01
```

```
>> impulse(Wd2,10)
```

```
>> bode(Wd2)
```

```
>> step(Wd2,10)
```

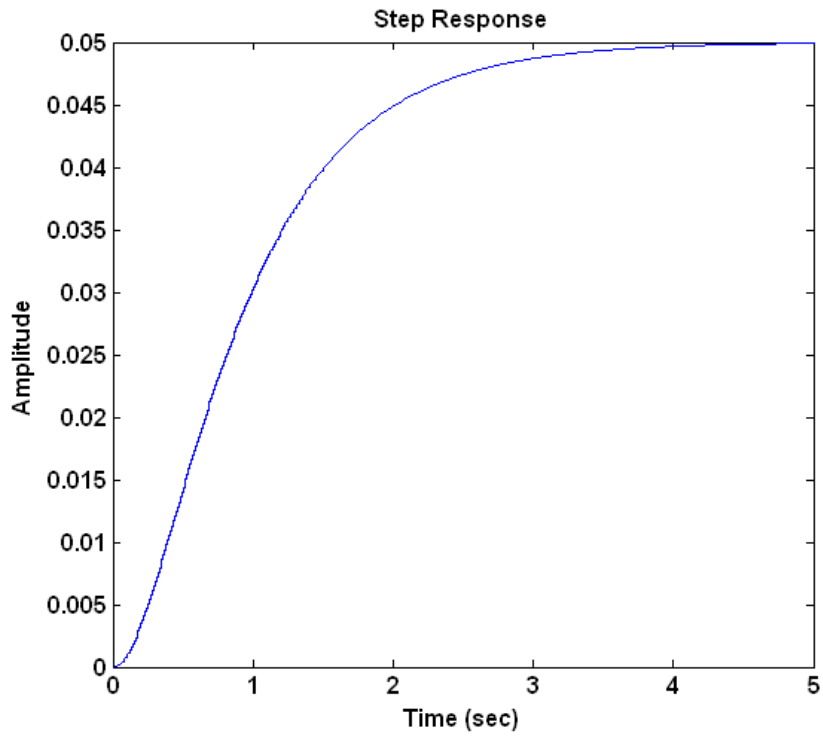


Рис. 7.5.12. Перехідна характеристика цифрової САУ для методу інваріантної імпульсної характеристики

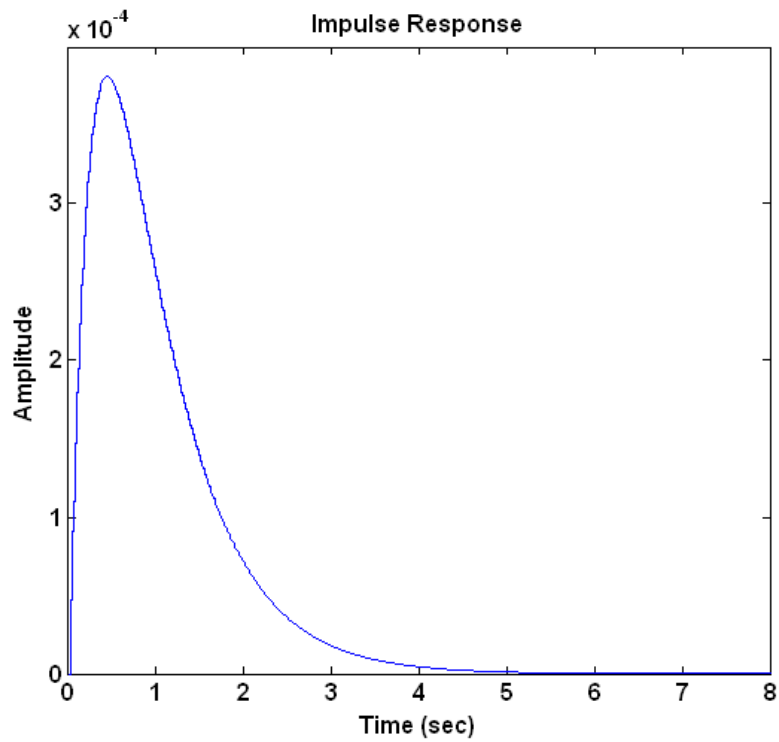


Рис. 7.5.13. Імпульсна характеристика цифрової САУ для методу інваріантної імпульсної характеристики

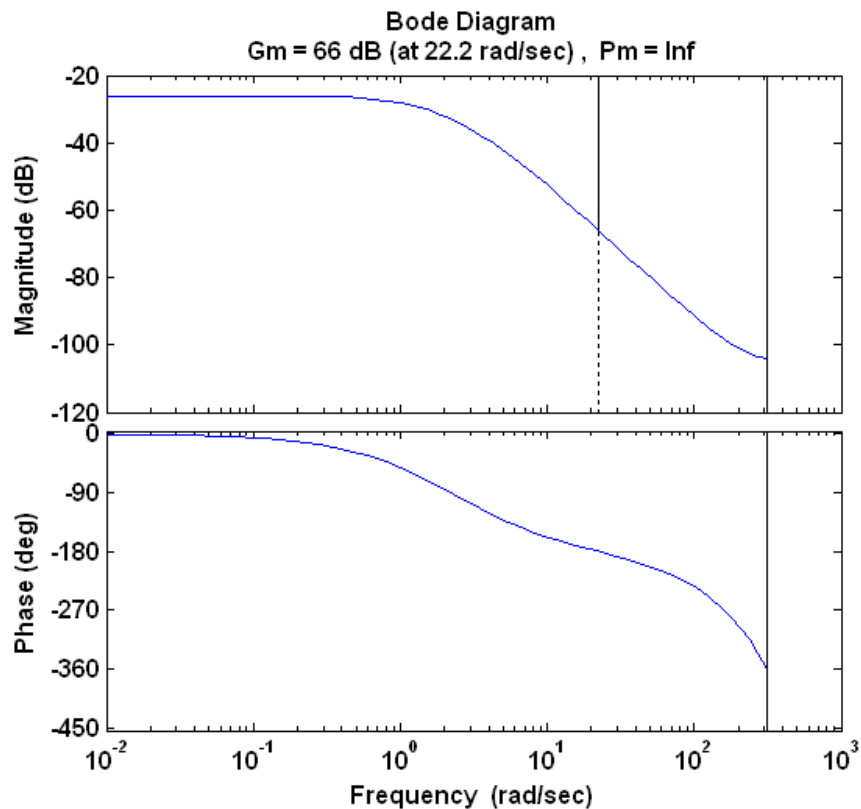


Рис. 7.5.14. Логарифмічна амплітудна характеристика цифрової САУ для методу інваріантної імпульсної характеристики

За допомогою програми MATLAB в пакеті Simulink цифрову САУ, перетворену різними методами, можна представити наступним чином (рис. 7.5.15-7.5.17).

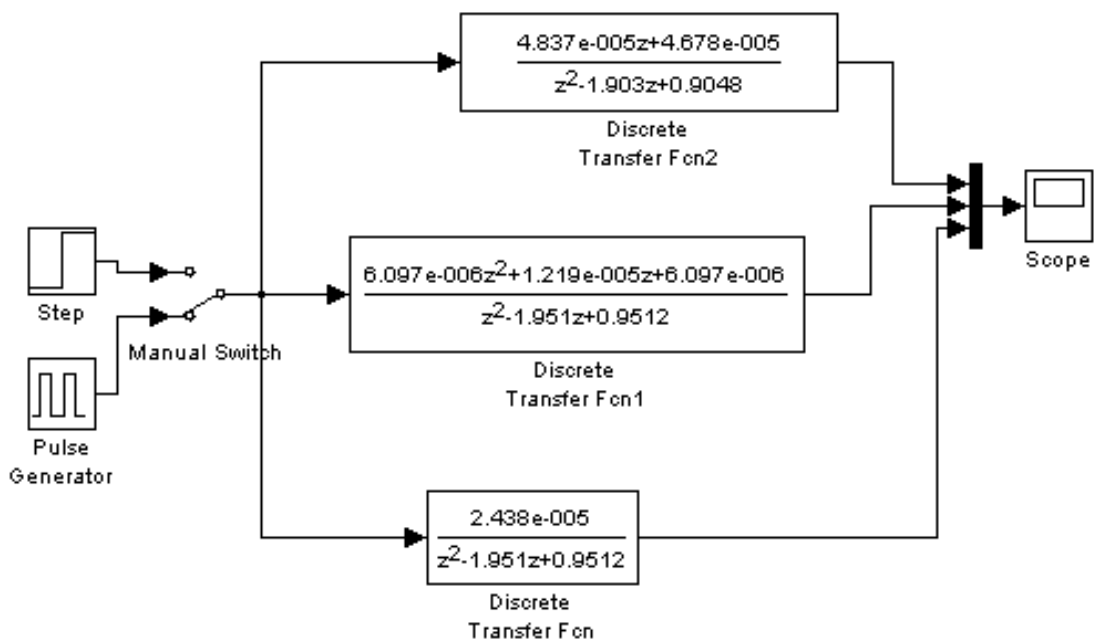


Рис. 7.5.15. Цифрові САУ, перетворені різними методами

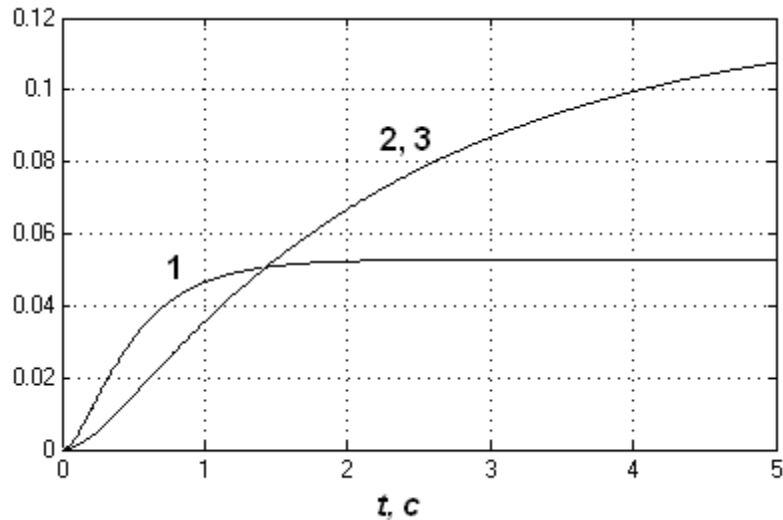


Рис. 7.5.16. Перехідні характеристики цифрових САУ, перетворених різними методами: 1- метод c2d, 2,3- методи bilinear та impinvar.

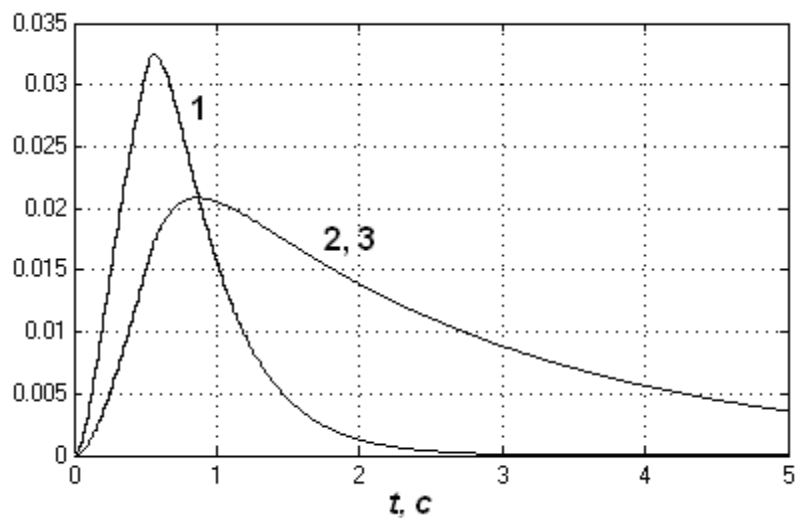


Рис. 7.5.17. Імпульсні характеристики цифрових САУ, перетворених різними методами: 1- метод c2d, 2,3- методи bilinear та impinvar.

7.5.2. Завдання для лабораторної роботи

1. Згідно свого варіанту обрати структурну схему системи автоматичного управління (додаток А) та параметри цієї схеми (додаток Б).

2. Перетворити безперервну САУ (отриману в лабораторній роботі 1, п.2.4) в цифрову систему (z – перетворення) різними методами та отримати перехідну, імпульсну, логарифмічно-амплітудну характеристики.

3. Про моделювати за допомогою пакету Simulink цифрові передаточні функції, отримані в п. 2, та отримати перехідну та імпульсну характеристику.

4. Оцінити на стійкість системи автоматичного управління за перехідною, імпульсною та логарифмічно-амплітудною характеристикою.

5. Порівняти графіки характеристик, отримані різними методами моделювання.

6. Порівняти безперервну та цифрову системи автоматичного керування шляхом аналізу отриманих графіків характеристик.

7.5.3. Зміст звіту

1. Назва та мета роботи.

2. Структурна схема системи автоматичного управління згідно варіанту свого завдання.

3. Результат перетворення безперервної системи в цифрову різними методами та графіки характеристик.

3. Результат моделювання цифрових САУ в Simulink та графіки характеристик.

4. Оцінювання на стійкість системи автоматичного управління.

5. Аналіз графіків характеристик, що отримані різними методами моделювання.

6. Аналіз та порівняння безперервної та дискретної цифрової системи автоматичного управління.

7. Висновки по роботі.

7.5.4. Контрольні питання

1. Дайте визначення дискретним системам.

2. Наведіть класифікацію дискретних систем.

3. Наведіть приклад зв'язку звичайної і дискретної (на основі z - перетворення) передаточних функцій.

4. В чому особливості частотних характеристик імпульсних та цифрових систем?

5. Як оцінюється стійкість імпульсних та цифрових систем?

6. Які особливості цифрових систем?

7. Як записуються передаточні функції цифрових систем?

7.6. Лабораторна робота 6. Перетворення математичних моделей цифрової системи управління в програмі MATLAB

Мета роботи: ознайомитися з методами введення та перетворення математичної моделі цифрової системи управління до середовища програми MATLAB за допомогою інструментальних пакетів Simulink та Control System Toolbox

7.6.1. Теоретичні відомості

Введення математичної моделі систем керування різними способами

Введення математичної моделі цифрової системи до середовища програми MATLAB (пакети Simulink та Control System Toolbox) має ті ж чотири формати, що і для безперервної системи:

- 1) у вигляді коефіцієнтів чисельників та знаменників передаточних функцій (поліномів);
- 2) в форматі матриць простору стану;
- 3) в форматі нулів, полюсів та коефіцієнтів передачі системи;
- 4) в форматі доданків простих дробів.

Для створення математичних моделей використовуються ті ж самі команди `tf`, `zpk`, `ss`, `frd` з додаванням в список їх аргументів часу вибірки (періоду квантування) T_s (в секундах):

```
sys = tf(num, den, Ts)
sys = zpk(z, p, k, Ts)
sys = ss(A, B, C, D, Ts)
```

Передаточна функція цифрової системи у вигляді поліномів

Одним з найпростіших форматів є введення математичної моделі у вигляді коефіцієнтів чисельників та знаменників передаточної функції. Передаточна функція записується в вигляді поліномів. Цей формат математичної моделі може бути представлений за допомогою пакета Control System Toolbox, наступним чином:

```
>>W=tf([1 -3 2],[1 2 -1 -2]);
>>Wd = c2d(W,0.01,'tustin');
>>impulse(W);
>>step(W);
>>bode(W);
```

Отримують імпульсну та перехідну характеристики, а також ЛАЧХ, ЛФЧХ.

Також даний формат можна представити за допомогою пакета Simulink (рис. 7.6.1).

В даному випадку на вхід подається одинична ступінчаста дія та одинична імпульсна дія, а на виході відповідно отримують перехідну та імпульсну характеристику.

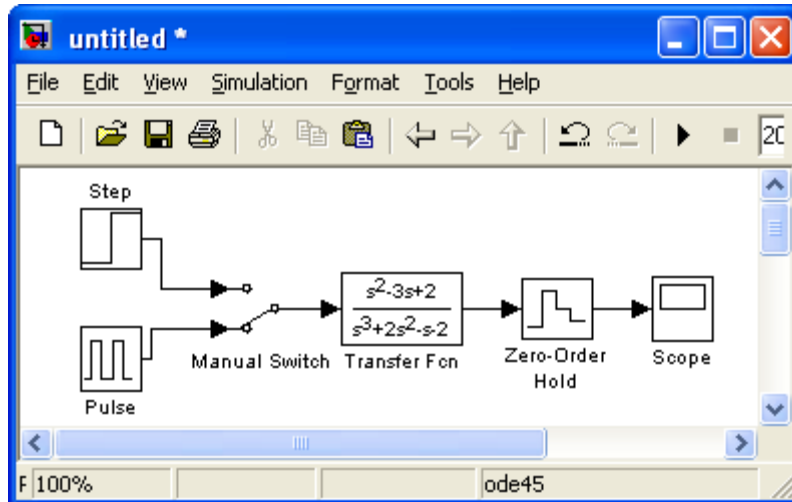


Рис. 7.6.1

Передаточна функція цифрової системи у просторі стану

Подібно рівнянь неперервної системи (7.2.4), (7.2.5), рівняння простору стану для дискретних систем мають вигляд:

$$\begin{aligned} X(n+1) &= AX(n) + BU(n), \\ Y(n) &= CX(n) + DU(n), \end{aligned} \quad (7.6.1)$$

де X – вектор стану,

U – вектор вхідних впливів

Y – вектор вихідних сигналів

A, B, C, D – матриці: системи, входу, виходу, обходу відповідно.

n – номер шага моделювання.

Даний формат математичної моделі може бути представлений за допомогою пакета Control System Toolbox, наступним чином:

```
>>[A, B, C, D] = ssdata (Wd)
>>Wdd=ss (A, B, C, D, 0.01) ;
>>impulse (Wdd) ;
>>step (Wdd) ;
>>bode (Wdd) ;
```

Отримують імпульсну та перехідну характеристики, а також ЛАЧХ, ЛФЧХ.

Також цей формат за допомогою пакета Simulink представлений на рис. 7.6.2.

В даному випадку на вхід подається одинична ступінчаста дія та одинична імпульсна дія, а на виході відповідно отримують перехідну та імпульсну характеристику.

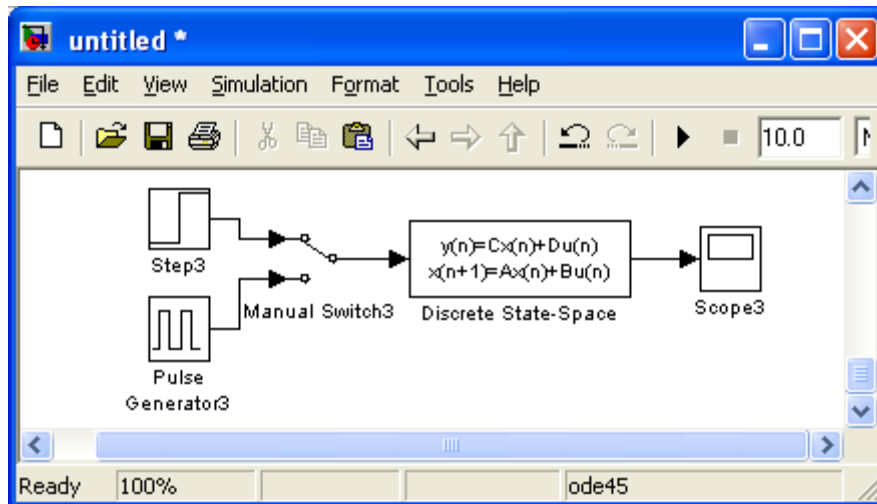


Рис. 7.6.2

Передаточна функція цифрової системи, заданої у вигляді нулів та полюсів

Розклавши чисельник та знаменник функції передачі, представлені у дискретній формі, на множники, отримаємо передаточну функцію, подібну до функції (2.6):

$$H(Z) = K \frac{Z(z)}{P(z)} = K \frac{(z - Z_1)(z - Z_2) \dots (z - Z_m)}{(z - P_1)(z - P_2) \dots (z - P_n)}, \quad (7.6.2)$$

де Z - вектор або матриця нулів передаточної функції,

P - вектор полюсів передаточної функції,

K - коефіцієнт передаточної функції, або вектор коефіцієнтів, якщо нулі передаточної функції задані матрицею. При цьому розмірність вектора K визначається числом рядків матриці нулів.

Кількість нулів не повинна перевищувати число полюсів передаточної функції.

Даний формат математичної моделі представлений за допомогою пакета Control System Toolbox наступним чином:

```
>>[z,p,k] = zpndata(Wd, 'v')
>>Wzp=zpk(z,p,k,0.01)
Zero/pole/gain:
0.0097522 (z-1.02) (z-1.01)
-----
(z-1.01) (z-0.99) (z-0.9802)
Sampling time: 0.01
>>step(Wzp)
>>impulse(Wzp)
>>bode(Wzp)
```

Також даний формат математичної моделі може бути представлений за допомогою пакета Simulink (рис. 7.6.3).

В даному випадку на вхід подається одинична ступінчаста дія та одинична імпульсна дія, а на виході відповідно отримують перехідну та імпульсну характеристики.

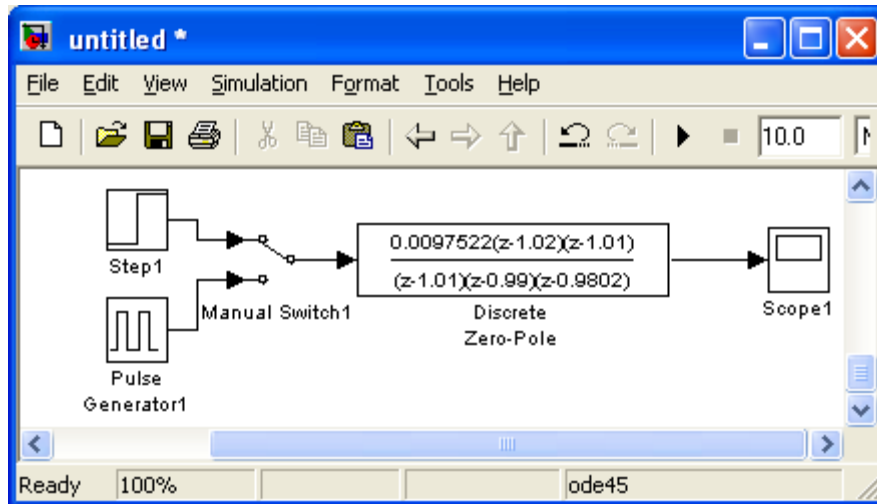


Рис. 7.6.3

Якщо нулі та полюси важко розрахувати, їх можна отримати, перетворюючи з одного формату в інший математичну модель (див. лабораторну роботу №2).

7.6.2. Завдання для лабораторної роботи

1. Згідно свого варіанту оберіть структурну схему системи автоматичного управління (додаток А) та параметри цієї схеми (додаток Б).

2. Введіть в ЕОМ передаточну функцію у дискретній формі у вигляді поліномів в пакеті Control System Toolbox та отримайте перехідну, імпульсну, ЛАЧХ та ЛФЧХ характеристики САУ.

3. Введіть в ЕОМ передаточну функцію у дискретній формі у вигляді поліномів за допомогою пакету Simulink та отримайте перехідну, імпульсну характеристики САУ.

4. Введіть в ЕОМ передаточну функцію у дискретній формі у вигляді простору стану в пакеті Control System Toolbox та отримайте перехідну, імпульсну, ЛАЧХ та ЛФЧХ характеристики САУ.

5. Введіть в ЕОМ передаточну функцію у дискретній формі у вигляді простору стану за допомогою пакету Simulink та отримайте перехідну та імпульсну характеристики САУ.

6. Введіть в ЕОМ передаточну функцію у дискретній формі у вигляді нулів та полюсів в пакеті Control System Toolbox та отримайте перехідну, імпульсну, ЛАЧХ та ЛФЧХ характеристики САУ.

7. Введіть в ЕОМ передаточну функцію у дискретній формі у вигляді нулів та полюсів за допомогою пакету Simulink та отримайте перехідну, імпульсну характеристики САУ.

8. Введіть в ЕОМ передаточну функцію у дискретній формі у вигляді простих дробів в пакеті Control System Toolbox та отримайте перехідну, імпульсну, ЛАЧХ та ЛФЧХ характеристики САУ.

9. Введіть в ЕОМ передаточну функцію у дискретній формі у вигляді простих дробів за допомогою пакету Simulink та отримайте перехідну, імпульсну характеристики САУ.

10. Оцініть на стійкість систему автоматичного управління за перехідною характеристикою та за ЛАЧХ, ЛФЧХ.

11. Порівняйте графіки характеристик, отримані за різними методами представлення передаточних функцій.

7.6.3. Зміст звіту

1. Назва та мета роботи.
2. Структурна схема системи автоматичного управління згідно варіанту свого завдання.
3. Результат моделювання САУ в Control System Toolbox та графіки характеристик.
4. Результат моделювання даної структурної схеми в Simulink та графіки характеристик.
5. Порівняння різних методів представлення передаточних функцій.
6. Оцінка на стійкість системи автоматичного управління.
7. Аналіз графіків характеристик, що отримані за різними методами моделювання.
8. Висновки по роботі.

7.6.4. Контрольні питання

1. Як ввести дискретну передаточну функцію у вигляді поліномів у пакеті Simulink та Control System Toolbox?
2. Як ввести дискретну передаточну функцію у вигляді простору стану в пакеті Simulink та Control System Toolbox?
3. Як ввести дискретну передаточну функцію у вигляді нулів та полюсів у пакеті Simulink та Control System Toolbox?
4. Як ввести дискретну передаточну функцію у вигляді простих дробів у пакеті Simulink та Control System Toolbox?
5. За допомогою яких команд дискретну передаточну функцію САУ можна перетворити з одного вигляду в інший?
6. За допомогою яких команд можна отримати дискретну перехідну, імпульсну, ЛАЧХ та ЛФЧХ характеристики?
7. Як впливає на моделювання САУ перетворення її моделі з одного формату в інший?

7.7. Лабораторна робота 7. Синтез регуляторів для цифрової системи управління часовими методами

Мета роботи: дослідити цифрові САУ часовими методами в програмі MATLAB (інструментальні пакети Control System Toolbox та Simulink), визначити показники якості перехідних процесів та вплив регуляторів на ці процеси.

7.7.1. Теоретичні відомості

7.7.1.1. Загальні відомості для дискретних ПІД-регуляторів

Пропорційно-інтегральний-диференціальний регулятор (ПІД-регулятор) входить у загальний контур управління та широко використовується в промислових системах управління. ПІД-регулятор розраховує значення "похибки" як різницю між вимірюваним значенням параметру процесу і бажаним значенням. Регулятор намагається звести до мінімуму похибки, регулюючи входи об'єкту управління.

Реалізація регулятора об'єкту (ПІД, ПІ, ПД, П або І типу) здійснюється за допомогою пакетів Simulink та Control System Toolbox. Параметри ПІД-регулятора можна налаштовувати вручну або автоматично.

Для ПІД-регулятора дискретного часу передаточна функція має вигляд:

$$C_{par} = P + Ia(z) + D \left[\frac{N}{1 + Nb(z)} \right], \quad (7.7.1)$$

$$C_{id} = P \left[1 + Ia(z) + D \frac{N}{1 + Nb(z)} \right], \quad (7.7.2)$$

де параметри a та b відповідно визначаються:

Метод чисельного інтегрування	інтегрування вперед методом Ейлера	інтегрування назад методом Ейлера	Метод трапецій
$a(z)$ метод ітерацій	$\frac{T}{z-1}$	$\frac{Tz}{z-1}$	$\frac{T}{2} \cdot \frac{z+1}{z-1}$
$b(z)$ метод фільтрації	$\frac{T}{z-1}$	$\frac{Tz}{z-1}$	$\frac{T}{2} \cdot \frac{z+1}{z-1}$

7.7.1.2. Реалізація ПІД-регуляторів за допомогою пакетів Simulink та Control System Toolbox

Нехай об'єкт керування має передаточну функцію в безперервному вигляді:

$$W(s) = \frac{10}{s^2 + s + 20}. \quad (7.7.3)$$

За допомогою пакету Control System Toolbox синтез ПД-регуляторів проводиться наступним чином:

```
>> num=10;
>> den=[1 1 20];
>> [numd,dend] = c2dm(num,den,0.01,'zoh');
>> Wz=tf(numd, dend, 0.01);
>> pidtool(Wz)
```

Функція `pidtool(Wz)` автоматично підбирає регулятор для системи, а також робить систему стійкою. Приклад застосування параметри цієї функції для системи з передаточною характеристикою Wz представлений на рис. 7.7.1.

```
>>pidtool(Wz)
```

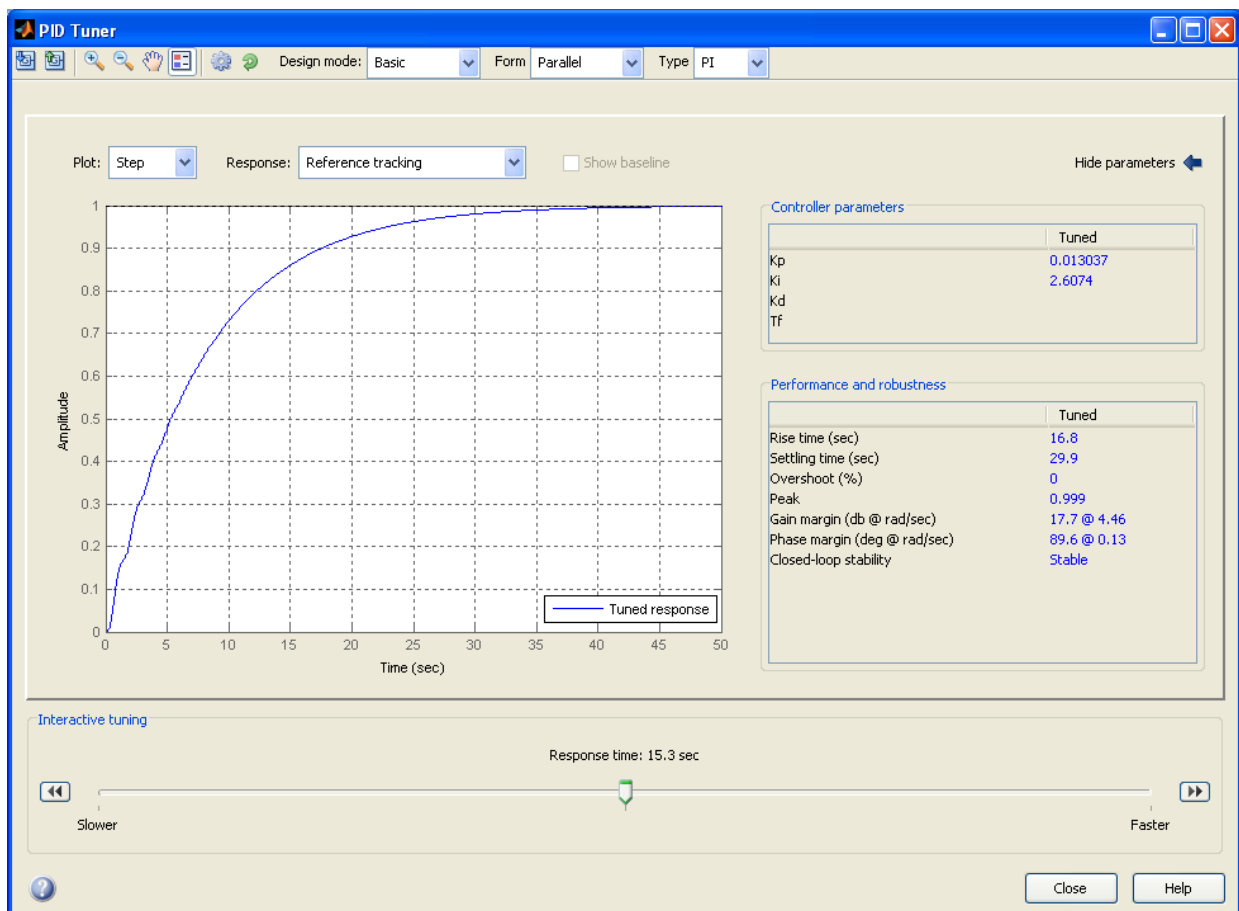


Рис. 7.7.1

Функція `pidtune(Wz, 'PID')` підбирає регулятор заданого типу для системи, робить її стійкою, а також покращує ЛАХ системи. Приклад застосування цієї функції для системи з передаточною характеристикою Wz представлений на рис. 7.7.2.

```
>> C=pidtune(Wz, 'PI')
Discrete-time PI controller in parallel form:
```

```

                                Ts
Kp + Ki * -----
                                z-1
with Kp = 0.013037, Ki = 2.6074, Ts = 0.01
>>F=feedback(series(WB,C),1,-1)
Transfer function:
6.496e-007 z^2 + 1.297e-006 z + 6.474e-007
-----
z^3 - 2.988 z^2 + 2.978 z - 0.99
Sampling time: 0.01
>> step(F)

```

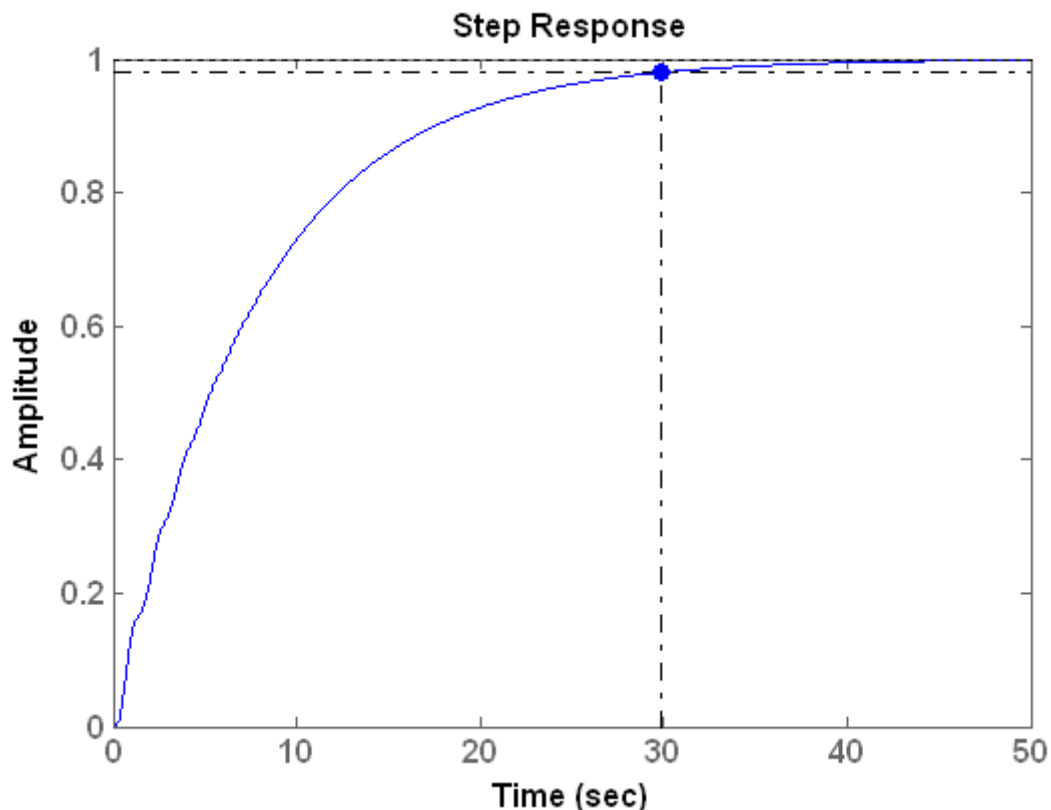


Рис. 7.7.2. Перехідна характеристика САУ з пропорційно-інтегральним регулятором

За допомогою пакету Simulink синтез ПІД-регуляторів проводиться наступним чином (рис. 7.7.3). Синтез за допомогою пакету Simulink може бути здійснений як автоматично, так и вручну, коли значення параметрів регуляторів заздалегідь відомі.

У випадку автоматичного синтезу при подвійному натисканні на ліву клавішу миші, що вказує на блок Discrete PID Controller, з'явиться вікно "Function Block Parameters: PID Controller". В цьому вікні можна обрати тип регулятора та натиснути кнопку Tune для автоматичного підбору параметрів регулятора.

У випадку ручного синтезу необхідно у вікно Function Block Parameters ввести значення параметрів регулятора.

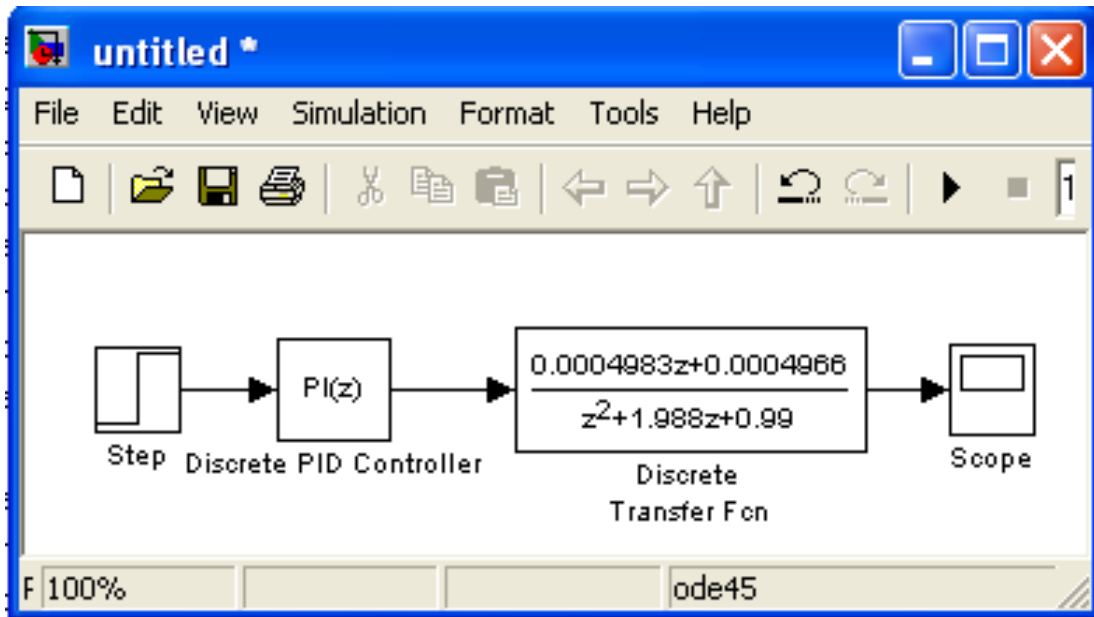


Рис. 7.7.3

Як видно з вище викладеного, в даному випадку застосовується ПІ-регулятор з параметрами $K_p = 0.01303\%$, $K_i = 2.6074$, часом встановлення $t_g = 29,9$, перерегулюванням $\sigma = 0\%$, похибка сталого режиму відсутня, система стійка.

7.7.2. Завдання для лабораторної роботи

1. Згідно свого варіанту обрати структурну схему системи автоматичного управління (додаток 1) та параметри цієї схеми (додаток 2).

2. Ввести передаточну функцію цифрової САУ у вигляді полінома за допомогою пакету Simulink та отримати відгук розімкнутого кола системи, прийняти проектне рішення щодо потрібних удосконалень цієї системи.

3. Ввести передаточну функцію цифровій системи у вигляді полінома в пакет Control System Toolbox та отримати відгук розімкнутого кола системи, прийняти проектне рішення щодо потрібних удосконалень цієї системи.

4. На основі отриманих графіків визначити час встановлення, перерегулювання, стійкість САУ.

5. Якщо цифрова система нестійка, то застосувати регулятор. Якщо стійка, то покращити перехідну характеристику системи шляхом застосування регулятора.

6. Промодельовати цифрову систему з регулятором за допомогою пакету Simulink та отримати перехідну характеристику системи.

7. Промодельовати цифрову систему з регулятором в пакеті Control System Toolbox та отримайте перехідну характеристику системи.

8. Порівняти отримані графіки перехідної характеристики цифрової системи з регулятором та без регулятора.

9. Порівняти отримані графіки перехідної характеристики цифрової системи з відповідними характеристиками неперервної системи.

7.7.3. Зміст звіту

1. Назва та мета роботи.

2. Структурна схема системи автоматичного управління згідно свого варіанту завдання.

3. Результат моделювання структурної схеми цифрової САУ в пакеті Simulink та графік отриманої перехідної характеристики системи без регулятора.

4. Результат моделювання цифрової САУ в пакеті Control System Toolbox та графік отриманої перехідної характеристики системи без регулятора.

5. Результат моделювання цифрової САУ в пакеті Simulink та графік отриманої перехідної характеристики системи з регулятором.

6. Результат виконання цифрової моделі отриманої в Control System Toolbox та графік отриманої перехідної характеристики з регулятором.

7. Розрахунок часу встановлення, перерегулювання, в системі без регулятора та з регулятором.

8. Визначення стійкості цифрової системи без регулятора та з регулятором.

9. Порівняння графіків перехідних характеристик цифрової системи автоматичного управління та безперервної системи автоматичного управління, покращених за допомогою регулятора, та отриманих різними методами моделювання.

9. Висновки по роботі.

7.7.4. Контрольні питання

1. Дайте визначення часової характеристики дискретної системи.

2. Що собою являють часові характеристики дискретних систем?

3. Як застосувати регулятор для покращення часових характеристик за допомогою пакету програм MATLAB?

4. Як застосувати регулятор для покращення часових характеристик САУ за допомогою пакету Control System Toolbox?

5. Який вплив кожного із елементів регулятора на якість перехідних процесів в САУ?

7.8. Лабораторна робота 8. Синтез регуляторів для цифрової системи управління частотними методами

Мета роботи: дослідити методи синтезу регуляторів для цифрової САУ частотними методами в програмі MATLAB (інструментальні пакети Control System Toolbox та Simulink), визначити показники якості перехідних процесів.

7.8.1. Теоретичні відомості

Для прогнозування якості роботи замкнутої системи на основі відгуків розімкнутої системи використовуються такі положення:

1. Якщо частота зрізу розімкнутої системи, менша від частоти, на якій ФЧХ досягає значення -180^0 ($\omega_c < \omega_{pc} = -180^0$), то замкнута система буде стійкою.

2. Коефіцієнт демпфування замкнутої системи 2-го порядку дорівнює запасу по фазі, що віднесений до 100.

3. Для систем 2-го порядку за межу смуги перепускання можна прийняти частоту власних незгасаючих коливань.

Використаємо вказані властивості при синтезі регулятора для системи, яка відповідає структурній схемі на рис. 7.8.1. В якості регулятора будемо використовувати різні типи регуляторів з передаточною функцією загального виду $G_c(p)$. Нехай об'єкт керування має передаточну функцію (7.7.1).

Проектуємий регулятор повинен задовольняти таким вимогам:

- забезпечувати нульову похибку в сталому режимі;
- перерегулювання не повинно перевищувати 40%;
- час усталення кривої перехідного процесу має бути $\leq 2c$.

Задачу синтезу регулятора можна вирішити графічним або чисельним методом. При використанні програми MATLAB графічний метод має явні переваги, тому будемо користуватися графічним методом синтезу регулятора. Спочатку побудуємо ЛАХ, для чого запишемо m-файл:

```
>> num=10;  
>> den=[1 1 20];  
>> [numd,dend] = c2dm(num,den,0.01,'zoh');  
>> Wz=tf(numd, dend, 0.01);  
>> Wzz=feedback(Wz,-1);  
>> margin(Wzz)
```

Графіки ЛАХ наведено на рис. 7.8.1.

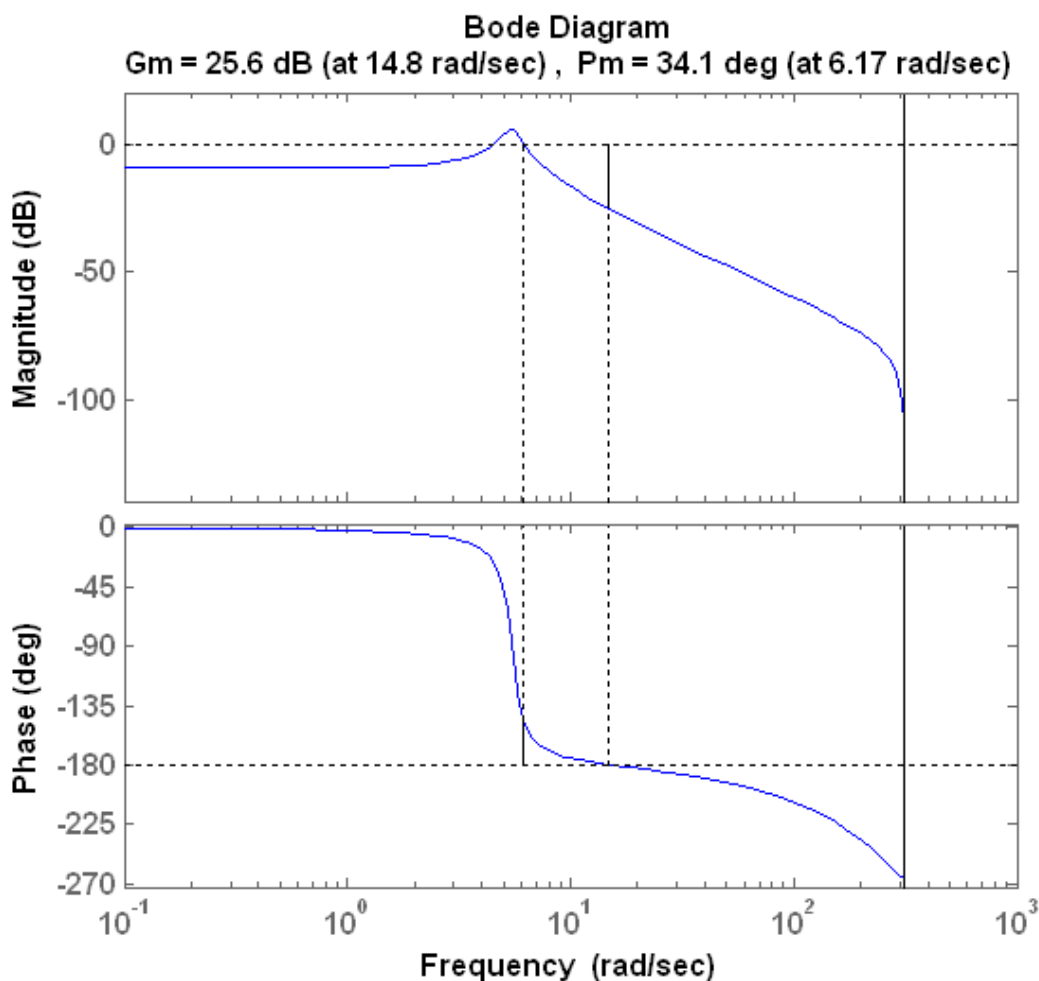


Рис. 7.8.1. Логарифмічно-амплітудна характеристика початкової замкнутої САУ

На основі графіків ЛАХ можна визначити деякі характеристики системи. Запас по амплітуді дорівнює 26,5 dB. Запас по фазі для цього об'єкта керування складає приблизно $\Delta\varphi(\omega_c)=34,1^\circ$. Це відповідає значенню коефіцієнта демпфування $\Delta\varphi(\omega_c)/100 = 34,1^\circ/100=0,341$.

З отриманих характеристик можна зробити висновок, що система стійка, тому потрібно покращити ці характеристики.

Отримані результати можна перевірити шляхом моделювання в програмі MATLAB відгуку замкнутої системи на ступеневий вхідний сигнал. Для цього записуємо *m*-файл:

```
>> step(Wzz)
```

Результатом розрахунків є графік, наведений на рис. 7.8.2.

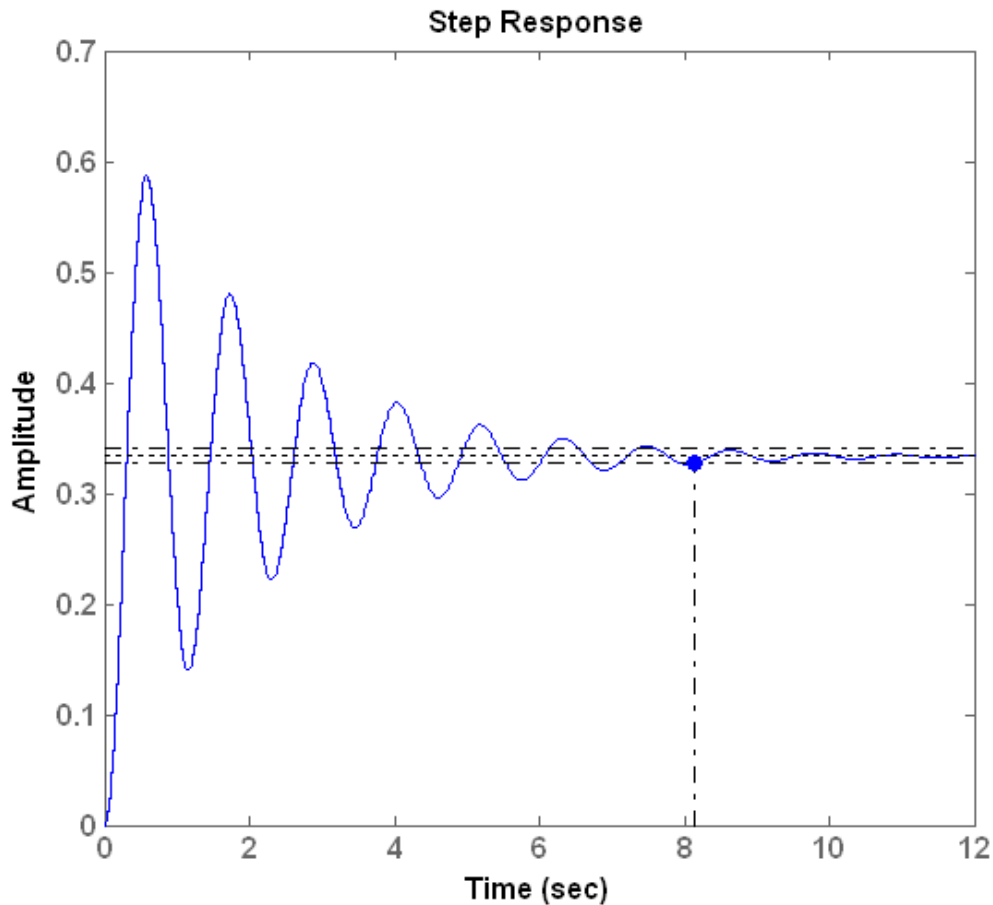


Рис. 7.8.2. Перехідна характеристика початкової замкнутої САУ

Таким чином, результати попереднього аналізу властивостей об'єкта керування підтверджуються: час встановлення кривої перехідного процесу становить близько 8,2 с, перегулювання $\sigma=45\%$, похибка сталого режиму також є неприпустимою (див. рис. 7.8.2).

Далі необхідно вибрати регулятор, який буде відповідати сформульованим раніше вимогам.

Функція `pidtool(Wz)` підбирає регулятор для системи, при цьому достатньо задати передаточну характеристику розімкненої системи, а функція автоматично знайде замкнену передаточну характеристику замкнутої системи та зробить її стійкою, покращить ЛАХ. Приклад застосування цієї функції для даної передаточної характеристики Wz представлений на рис. 7.8.3.

```
>>pidtool(Wz)
```

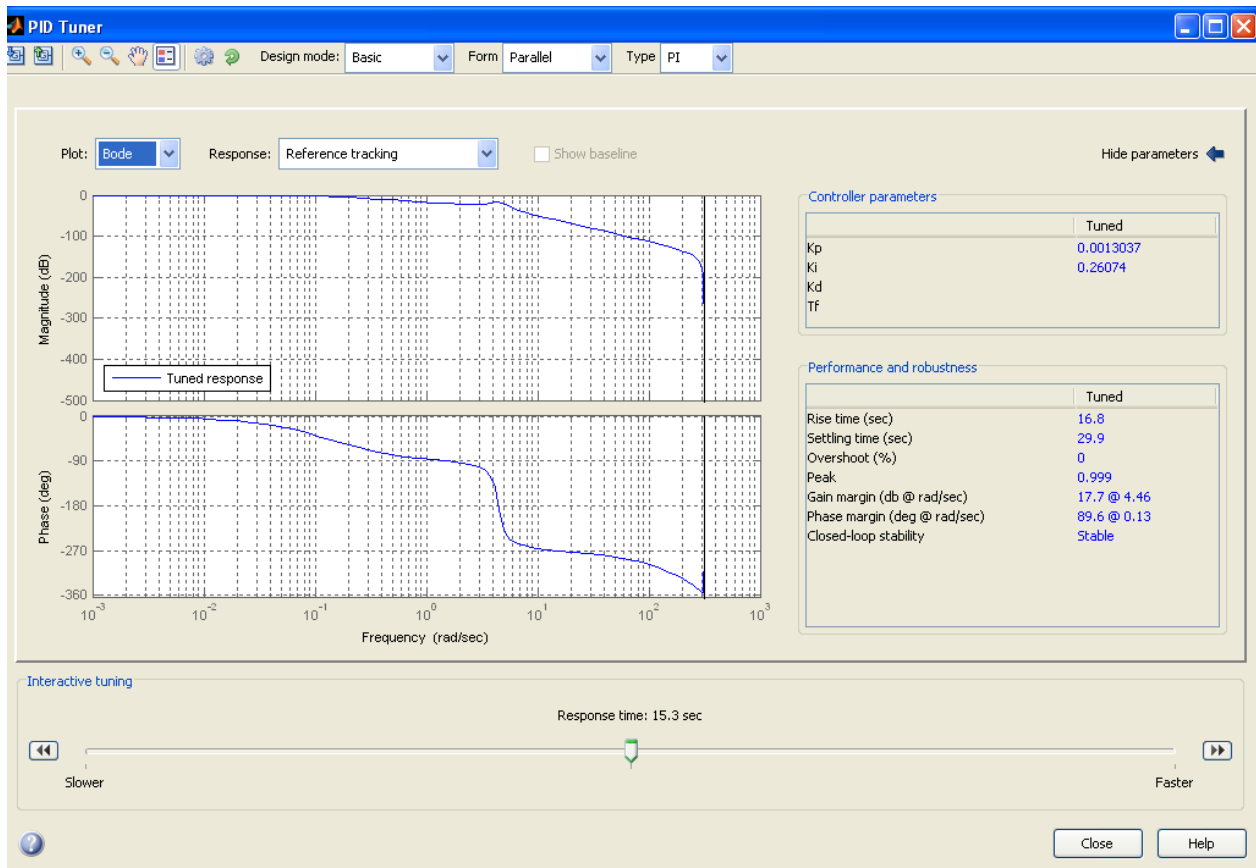


Рис. 7.8.3

З рис.8.3 видно, що в даному випадку застосовується ПІ-регулятор з параметрами $K_p = 0.0013037$, $K_i = 0.26074$. Запас по фазі складає $89,6^\circ$, запас по амплітуді $-17,7\text{dB}$, час встановлення $t_g = 29,9$, перерегулювання $\sigma = 0\%$, похибка сталого режиму відсутня, система стійка.

Функція `pidtune(Wzz, 'PID')` підбирає регулятор заданого типу, робить САУ стійкою та покращує її ЛАХ. Приклад застосування цієї функції для САУ з передаточною функцією `Wzz` представлений на рис. 7.8.4.

```
>>C=pidtune(Wz, 'PI')
```

Discrete-time PI controller in parallel form:

$$K_p + K_i * \frac{T_s}{z-1}$$

with $K_p = 0.0013037$, $K_i = 0.26074$, $T_s = 0.01$

```
>> F=feedback(series(Wz,C),1,-1)
```

Transfer function:

$$\frac{6.496e-007 z^2 + 1.297e-006 z + 6.474e-007}{z^3 - 2.988 z^2 + 2.978 z - 0.99}$$

```
Sampling time: 0.01
>> margin(F)
```

При застосуванні функції `pidtune` отримуємо запас по фазі 16,4 dB та запас по амплітуді -180°, що відрізняються від результатів функції `pidtool`. Графіки перехідного процесу навпаки співпадають. Розбіжності виникають через різні методи обчислення в самих функціях.

Таким чином, спрощений варіант синтезу регулятора полягає в підборі структури та параметрів регулятора, які повинні забезпечити визначені в технічному завданні на проектування показники динамічних властивостей проектуємої системи.

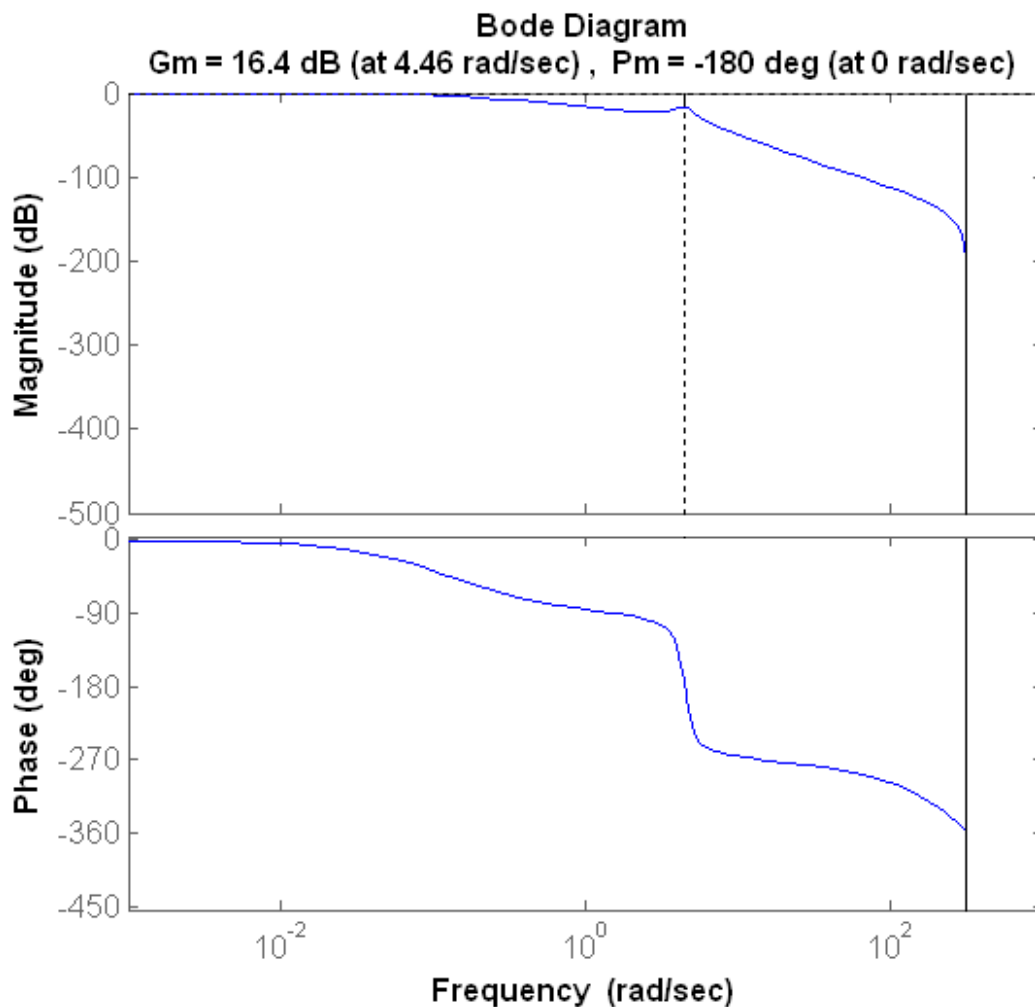


Рис. 7.8.4. Логарифмічна-амплітудна характеристика пропорційно-інтегрального регулятора

7.8.2. Завдання для лабораторної роботи

1. Згідно свого варіанту обрати структурну схему системи автоматичного управління (додаток А).
2. Параметри цієї схеми взяти з таблиці (додаток Б).
3. Ввести передаточну функцію у вигляді полінома в пакет Control System Toolbox та отримати ЛАХ системи.
4. На основі отриманих графіків визначити смугу перепускання, запас по фазі, перерегулювання, час встановлення, стійкість системи.
5. Якщо система нестійка, то застосувати регулятор. Якщо стійка, то покращити ЛАХ (задачу синтезу регулятора вирішити графічним методом).
6. Для оцінки отриманих параметрів регулятора виконати моделювання цифрової системи і отримати перехідну характеристику замкненої системи
7. Порівняйте отримані графіки ЛАХ системи з регулятором та без регулятора.

7.8.3. Зміст звіту

1. Назва та мета роботи.
2. Структурна схема системи автоматичного управління згідно свого завдання.
3. Результат моделювання в пакеті Control System Toolbox та графік отриманої ЛАХ системи без регулятора.
4. Результат моделювання в пакеті Control System Toolbox та графік отриманої ЛАХ системи з регулятором.
6. Розрахунок смуги перепускання, запасу по фазі, перерегулювання, часу встановлення, визначення стійкості системи.
7. Аналіз графіків ЛАХ системи автоматичного управління, покращених за допомогою регулятора.
8. Висновки по роботі.

7.8.4. Контрольні питання

1. Який вплив кожного з елементів регулятора на якість перехідних процесів?
2. Що собою являють частотні характеристики цифрових систем?
3. Дайте визначення кожної з частотних характеристик цифрових систем.
4. Як застосувати регулятор для покращення частотних характеристик за допомогою пакету програм MATLAB?
5. Як застосувати регулятор в програмі MATLAB для покращення частотних характеристик САУ?

7.9. Лабораторна робота 9. Застосування фільтра Калмана для вимірювання поточних координат та параметрів руху об'єктів управління

Мета роботи: ознайомитися з можливостями підвищення точності визначення координат та параметрів руху об'єктів в системах управління.

7.9.1. Теоретичні відомості

7.9.1.1. Основні відомості з теорії Калмановської оптимальної фільтрації

Проблема оптимальної фільтрації виникла на початку 40-х років у зв'язку з підвищенням вимог до точності сигналів керування.

Спочатку вона розв'язувалась шляхом побудови фільтрів А.Н. Колмогорова для дискретних випадкових сигналів і Н. Вінера – для неперервних стаціонарних випадкових процесів. При цьому структура фільтра визначалась в результаті розв'язання інтегрального рівняння Вінера-Хопфа для стаціонарних режимів роботи об'єкта, що значно обмежувало можливості застосування.

Теорія лінійної оптимальної фільтрації, розроблена Калманом і Бьюсі в 1960 році, дозволяє визначити структуру фільтра в часовому просторі, використовуючи концепцію “простору стану”.

Успіхи цієї теорії у порівнянні з ідеями вінеровської фільтрації пояснюються тим, що структуру оптимального фільтра одержуємо в результаті розв'язання звичайних диференціальних рівнянь, що спростило інженерні розробки, а також тим, що з'явилась можливість одержання рекурентної системи алгоритмів оптимізації структури фільтра, зручних для реалізації на ЦОМ.

Методи фільтра Калмана легко поширюються на задачі визначення недоступних безпосередньому вимірюванню координат об'єкта, що дозволяє одержати їх оптимальну оцінку в умовах дії перешкод.

Розглянемо структурну схему (рис. 7.9.1), де

ОУ – об'єкт управління;

ФК – фільтр Калмана.

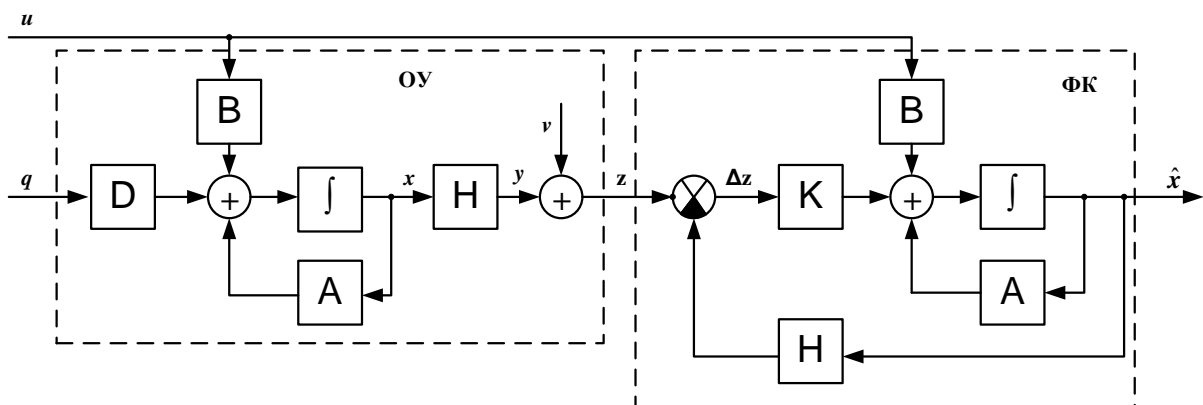


Рис. 7.9.1. Структурна схема отримання оцінок координат ОУ

Задача полягає в побудові такого фільтра Калмана, який відновлював би вектор стану об'єкта x з максимальною точністю. Фізично задача фільтрації полягає, по можливості, в найбільш повному подоланні перешкод вимірювання $v(t)$.

7.9.1.2. Оптимальне послідовне оцінювання параметрів незбурюваної детермінованої траєкторії руху ОУ

Нехай вимірюється послідовність значень параметрів, що має вигляд

$$U_i = H v_i + \Delta U_i, \quad (7.9.1)$$

де U_i - l -мірний вектор координат об'єкта управління, що спостерігаються; v_i - $(s \times 1)$ -мірний вектор параметрів траєкторії руху, що оцінюється; H - матриця розмірності $(l \times s)$, що встановлює однозначну відповідність між оцінюваними параметрами та координатами вимірювання; ΔU_i - l -мірний вектор помилок вимірювання координат; послідовність цих векторів вважається некорельованою випадковою послідовністю з математичним очікуванням, що дорівнює нулю, та з відомою кореляційною матрицею.

Вважається також, що між попередніми і наступними значеннями параметрів є однозначний зв'язок, так що значення параметрів в двох сусідніх процедурах вимірювання пов'язані співвідношенням

$$v_{i+1} = F v_i, \quad (7.9.2)$$

де F – матриця, розмірності $s \times s$.

Задача оцінювання полягає в отриманні рекурентних виразів для оцінки значень параметрів руху ОУ за результатами вимірювань з похибками.

7.9.1.3. Формульна схема оптимального алгоритму послідовного оцінювання векторного параметру руху ОУ.

Як відомо із загальної теорії фільтрації, найкращий шлях рішення задачі послідовного оцінювання та згладжування параметрів складається в визначенні апостеріорної ймовірності фільтруємих параметрів, так як вона містить всю інформацію, що отримана з апіорних джерел та результатів вимірювань. Знаючи апостеріорну щільність ймовірності, можна отримати різні оцінки цікавлячих нас параметрів, в тому числі оцінки, що відповідає максимуму функції апостеріорної щільності. Останні, як відомо, називається оцінками, оптимальними по критерію максимуму апостеріорної ймовірності. Саме в цьому сенсі і розуміється оптимальна фільтрація в подальшому. Розглянемо в загальному вигляді задачу послідовного оцінювання та згладжування вектора параметрів траєкторії руху ОУ. При поліноміальному представленні траєкторії складовими цього вектору є: координати, швидкість зміни координат, прискорення по координатам тощо.

Вектор згладжених параметрів будемо позначати через \hat{v}_n з індексом n , що вказує його прив'язку до часу t_n . Порядок розташування складових вектору: координата, швидкість, прискорення ОУ. Одночасно з послідовним уточненням вектору оцінюваних параметрів будемо формувати також послідовну кореляційну матрицю помилок оцінки цих параметрів. Матриця Ψ_n визначає точнісні характеристики згадуваних параметрів на момент часу t_n та має розмірність $s \times s$.

Отже, нехай отримано згладжене значення \hat{v}_{n-1} вектора параметрів \hat{v}_{n-1} траєкторії цілі за результатами $n-1$ попередніх вимірювань її координат. Розподіл вектора \hat{v}_{n-1} приймається нормальним з математичним очікуванням v_{n-1} та кореляційною матрицею Ψ_{n-1} .

Вектор параметрів v_{n-1} екстраполюється на момент наступного (n -го) вимірювання. Екстрапольоване значення вектору параметрів обчислюється у відповідності із співвідношенням

$$\hat{v}_{ne} = F_e \hat{v}_{n-1}, \quad (7.9.3)$$

де F_e – відомий оператор екстраполяції параметрів. Конкретний вид оператора F_e визначається моделлю траєкторії руху ОУ.

Нехай, наприклад, вектор оцінюваних параметрів траєкторії в момент часу t_{n-1} має вигляд

$$\hat{v}_{n-1} = \left\| \hat{r}_{n-1}, \hat{\dot{r}}_{n-1}, \hat{\ddot{r}}_{n-1} \right\|^T, \quad (7.9.4)$$

що відповідає представленню окремо взятої координати ОУ у вигляді поліному другого ступеня.

В цьому випадку оператор екстраполяції параметрів на час $\tau_{\hat{a}} = (t_n - t_{n-1})$ має вигляд

$$F_e = \left\| \begin{array}{ccc} 1 & \tau_e & \frac{\tau_e^2}{2} \\ 0 & 1 & \tau_e \\ 0 & 0 & 1 \end{array} \right\|, \quad (7.9.5)$$

Тоді вираз (7.9.3) записується у вигляді

$$\hat{v}_{ne} = \left\| \begin{array}{c} \hat{r}_{ne} \\ \hat{\dot{r}}_{ne} \\ \hat{\ddot{r}}_{ne} \end{array} \right\| = \left\| \begin{array}{ccc} 1 & \tau_e & \frac{\tau_e^2}{2} \\ 0 & 1 & \tau_e \\ 0 & 0 & 1 \end{array} \right\| \cdot \left\| \begin{array}{c} \hat{r}_{n-1} \\ \hat{\dot{r}}_{n-1} \\ \hat{\ddot{r}}_{n-1} \end{array} \right\| \quad (7.9.6)$$

Кореляційна матриця Ψ_{n-1} помилок оцінки параметрів руху по результатам $(n-1)$ попередніх вимірювань також перераховується (екстраполюється) на момент наступного вимірювання, тобто на час τ_e .

Матричний оператор перерахунку кореляційної матриці помилок оцінки параметрів до моменту часу чергового вимірювання координат зазвичай співпадає з оператором F_e . Однак у деяких, практично важливих випадках, цей оператор може відрізнятись від F_e , тому для нього вводиться позначення Φ .

Матриця помилок оцінки екстрапольованих параметрів обчислюється наступним чином. Відповідно до формули (7.9.3) для вектору помилок екстраполяції параметрів в n -м огляді можна записати

$$\Delta v_{ne} = \Phi \Delta \hat{v}_{n-1}. \quad (7.9.7)$$

За визначенням,

$$\Psi_{ne} = M[\Delta \hat{v}_{ne} \Delta \hat{v}_{ne}^T], \quad (7.9.8)$$

де M – знак математичного очікування.

Маючи на увазі, що

$$\Delta \hat{v}_{ne}^T = \Delta \hat{v}_{n-1}^T \Phi^T, \quad (7.9.9)$$

отримуємо

$$\Psi_{ne} = \Phi \cdot M[\Delta \hat{v}_{n-1} \Delta \hat{v}_{n-1}^T] \Phi^T. \quad (7.9.10)$$

Замінюючи

$$M[\Delta \hat{v}_{n-1} \Delta \hat{v}_{n-1}^T] = \Psi_{n-1} \quad (7.9.11)$$

отримуємо остаточно

$$\Psi_{ne} = \Phi \Psi_{n-1} \Phi^T. \quad (7.9.12)$$

З урахуванням припущення про лінійність оператору екстраполяції, закон розподілу вектору екстрапольованих параметрів буде нормальним. У векторно-матричній формі відповідна щільність ймовірності записується наступним чином:

$$\omega(\hat{v}_{ne}) = C_1 \exp\left[-\frac{1}{2}(\hat{v}_{ne} - v_n)^T \Psi_{ne}^{-1}(\hat{v}_{ne} - v_n)\right], \quad (7.9.13)$$

де \hat{v}_n – вектор істинних значень параметрів у момент t_n . Щільність ймовірності (7.9.13) є апіорної щільністю ймовірності для вектору оцінюваних параметрів перед черговим (n -м) вимірюванням.

У момент часу t_n проводиться черговий вимір поточних координат ОУ. Вектор вимірюваних значень координат позначається через U_n . У загальному випадку $U_n = \|r_n, \beta_n, \varepsilon_n\|^T$.

Передбачається, що помилки вимірювання координат підпорядковані нормальному закону розподілу і некорельовані в суміжних вимірюваннях. Тому умовна щільність ймовірності вибірки вимірюваних значень координат має вигляд

$$\omega(U_n | v_n) = C_2 \exp\left[-\frac{1}{2}(U_n - H v_n)^T \Xi^{-1}(U_n - H v_n)\right], \quad (7.9.14)$$

де Ξ^{-1} – зворотна кореляційна матриця помилок вимірювання, яка внаслідок незалежності вимірюваних координат має вигляд

$$\Xi^{-1} = \begin{vmatrix} 1/\sigma_{r_n}^2 & 0 & 0 \\ 0 & 1/\sigma_{\beta_n}^2 & 0 \\ 0 & 0 & 1/\sigma_{\varepsilon_n}^2 \end{vmatrix}; \quad (7.9.15)$$

H – лінійний оператор відповідності оцінюваних параметрів і вимірюваних координат.

Наприклад, якщо вимірюються координати r_n, β_n і ε_n , а оцінюються параметри $\hat{r}_n, \hat{\beta}_n, \hat{\varepsilon}_n$, то оператор H має вигляд прямокутної матриці порядку 3×6 :

$$H = \begin{matrix} & \hat{r}_n & \hat{\beta}_n & \hat{\varepsilon}_n & & \\ \begin{matrix} r_n \\ \beta_n \\ \varepsilon_n \end{matrix} & \begin{vmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{vmatrix} \end{matrix}. \quad (7.9.16)$$

Добуток Hv_n у формулі (7.9.14) представляє собою вектор істинних значень вимірюваних координат у момент часу t_n .

Внаслідок відсутності кореляції помилок вимірювання апостеріорний розподіл для вектора оцінок параметрів після n -го вимірювання координат визначається по формулі Байеса

$$\omega(\hat{v}_n | U_n) = C_3 \omega(\hat{v}_{ne}) \omega(U_n | v_n), \quad (7.9.17)$$

де C_3 - нормуючий множник, визначаючий масштаб ламаної $\omega(\hat{v}_n | U_n)$ таким чином, щоб площа під цією ламаною була рівна одиниці.

Внаслідок нормальності складових розподілів апостеріорний розподіл (7.9.17) є нормальним. Відповідна щільність ймовірності записується у вигляді

$$\omega(\hat{v}_n | U_n) = C_4 \exp \left[-\frac{1}{2} (\hat{v}_n - v_n)^T \Psi_n^{-1} (\hat{v}_n - v_n) \right]. \quad (7.9.18)$$

де \hat{v}_n – вектор згладжених параметрів за результатами n вимірювань координат; Ψ_n - матриця помилок оцінки згладжених параметрів.

Для нормального розподілу $\max \omega(\hat{v}_n | U_n)$ співпадає з математичним очікуванням вектору оцінюваних параметрів. З цього випливає, що задача оцінки параметрів по максимуму апостеріорної ймовірності зводиться у нашому випадку до знаходження параметрів \hat{v}_n і Ψ_n , у виразі (7.9.15).

Використовуючи вирази (7.9.13), (7.9.14) та (7.9.18) для щільності ймовірності, що входить в формулу (7.9.17), після логарифмування отримуємо:

$$\begin{aligned}
(\hat{v}_n - v_n)^T \Psi_n^{-1} (\hat{v}_n - v_n) &= (\hat{v}_{ne} - v_n) \Psi_{ne}^{-1} (\hat{v}_{ne} - v_n) + \\
&+ (U_n - H v_n)^T \Xi^{-1} (U_n - H v_n) + const.
\end{aligned}
\tag{7.9.19}$$

Останнє рівняння є вихідним для знаходження вектору \hat{v}_n та матриці Ψ_n . Так, виділення доданків, що представляють собою квадратичні форми для вектора \hat{v}_n , дає

$$v_n^T \Psi_n^{-1} v_n = v_n^T \Psi_{ne}^{-1} v_n + v_n^T H^T \Xi^{-1} H v_n \tag{7.9.20}$$

З цього рівняння отримуємо

$$\Psi_n^{-1} = \Psi_{ne}^{-1} H^T \Xi^{-1} H \tag{7.9.21}$$

Операції обчислення матриць у виразі (7.9.21) призводять до наступного остаточного результату:

$$\Psi_n = \Psi_{ne} - \Psi_{ne} H^T (H \Psi_{ne} H^T + \Xi)^{-1} H \Psi_{ne} \tag{7.9.22}$$

Порівнюючи в рівнянні (7.9.19) квадратичні форми, що містяться ліворуч, отримуємо:

$$v_n^T \Psi_n^{-1} \hat{v}_n = v_n^T \Psi_{ne}^{-1} \hat{v}_{ne} + v_n^T H^T \Xi^{-1} U_n. \tag{7.9.23}$$

З цього рівняння знаходимо

$$\hat{v}_n = \Psi_n [\Psi_{ne}^{-1} \hat{v}_{ne} + H^T \Xi^{-1} U_n], \tag{7.9.24}$$

Після елементарних перетворень з урахуванням виразів (7.9.3) і (7.9.21) отримуємо остаточно:

$$\hat{v}_n = \hat{v}_{ne} + \Psi_n H^T \Xi^{-1} (U_n - H \hat{v}_{ne}). \tag{7.9.25}$$

У відповідність з виразом (7.9.25) вектор згладжених значень параметрів за результатами n вимірювань координат отримуємо як суму вектора екстрапольованого на момент n -го вимірювання параметрів і зваженої з деяким коефіцієнтом згладжування неузгодженості між виміряними і екстрапольованими значеннями координат.

Нижче наведено основні співвідношення оптимального алгоритму послідовного оцінювання та згладжування параметрів траєкторії руху ОУ, що записані в порядку виконання обчислювальних операцій:

- 1) $\hat{v}_{ne} = F_e \hat{v}_{n-1}$; (7.9.26)
- 2) $\Psi_{ne} = \Phi \Psi_{n-1} \Phi^T$;
- 3) $\Psi_n = \Psi_{ne} - \Psi_{ne} H^T (H \Psi_{ne} H^T + \Xi)^{-1} H \Psi_{ne}$;
- 4) $\hat{v}_n = \hat{v}_{ne} + \Psi_n H^T \Xi^{-1} (U_n - H \hat{v}_{ne})$,

де U_{ne} - вектор екстрапольованих значень координат. Система рівнянь (7.9.26) фактично є рівнянням фільтра Калмана.

7.9.1.4. Реалізація фільтра Калмана в програмі MATLAB

Для синтезу фільтра Калмана в програмі MATLAB призначені наступні функції інструментального пакету Control System Toolbox:

- для неперервних систем:

```
[KEST, L, P] = KALMAN(SYS, QN, RN, NN)
```

```
[KEST, L, P] = kalman(sys, Qn, Rn, Nn, sensors,  
known)
```

- для дискретних моделей:

```
[KEST, L, P, M, Z] = kalman(sys, Qn, Rn, Nn)
```

- для синтезу дискретного фільтра Калмана для безперервних систем:

```
[KEST, L, P, M, Z] = kalmz(sys, Qn, Rn, Nn)
```

Матриця оцінки параметрів фільтра Кальмана KEST для безперервного або дискретного часу моделюється в просторі стану за моделлю SYS.

Наведемо приклад розрахунку фільтра Калмана для системи з передаточною функцією

$$W(s) = \frac{100}{s^2 + s + 100},$$

та параметрами $Q = 0.1$, $R = 0.05$.

1. Розрахунок параметрів фільтра Калмана

```
>> Sys1=ss(tf(100, [1 1 100]))
```

```
a =
```

	x1	x2
x1	-1	-12.5
x2	8	0

```
b =
```

	u1	u2
x1	4	4
x2	0	0

```
c =
```

	x1	x2
y1	0	3.125

```
d =
```

	u1	u2
y1	0	0

Continuous-time model.

```
>> [A,B,C,D]=ssdata(Sys1)
```

```
A =
```

```
-1.0000  -12.5000  
 8.0000      0
```

```
B =
```

```
 4  
 0
```

```
C =
```

```
 0  3.1250
```

```
D =
```

```

0
>> [A,B,C,D]=ssdata(Sys1)
A =
-1.0000  -12.5000
 8.0000      0
B =
 4
 0
C =
 0  3.1250
D =
 0
>> P=ss(A, [B B], C, [D D])%моделювання системи
управління зі спостерігачем
a =
      x1      x2
x1    -1  -12.5
x2     8     0
b =
      u1  u2
x1     4   4
x2     0   0
c =
      x1      x2
y1     0  3.125
d =
      u1  u2
y1     0   0
Continuous-time model.Continuous-time model.
>> Kest=kalman(P,1,0.01) %моделювання фільтра
Калмана
a =
      x1_e      x2_e
x1_e     -1  -20.26
x2_e      8  -11.14
b =
      u1      y1
x1_e     4  2.483
x2_e     0  3.56
c =
      x1_e      x2_e
y1_e     0  3.125
x1_e     1     0
x2_e     0     1

```



```

d =
      u1  y1
  y1_e   0   0
  x1_e   0   0
  x2_e   0   0
I/O groups:
Group name      I/O Channel(s)
KnownInput      I      1
Measurement     I      2
OutputEstimate  O      1
StateEstimate   O      2,3
Continuous-time model.

```

2. Входом системи управління є вектор U , точним виходом системи – Y , YN – вихід з шумом, оцінка виходу системи у фільтрі Калмана – YK . Нижче наведено фрагмент програми моделювання та отримані результати (рис. 7.9.2 – 7.9.3).

```

U=ones(1000,1); % вхідний сигнал одиничний
ступеневий вплив 1000 відліків
T=linspace(0,10,1000)'; % вектор часу від 0 до 10 с
1000 відліків
Y=lsim(Sys1,U,T); % моделювання точного виходу
системи
YN=Y+0.05*randn(1000,1); % вихід с шумом СКО 0.05
figure; plot(T,U,'-g',T,Y,'-b',T,YN,'-r'); grid on;
% графк вхідного і вихідного сигналів системи
xlabel('Час, с'); ylabel('Амплітуда вихідного
сигналу, Вт');
[A,B,C,D]=ssdata(Sys1); % знаходимо значення
матриць системи
P=ss(A, [B B], C, [D D]) % будуємо систему зі
спостерігачем
Kest=kalman(P,0.1,0.05); % будуємо фільтр Калмана
YK=lsim(Kest,[U YN],T); % розраховуємо оцінку
виходу системи у фільтрі Калмана
figure; plot(T,U,'-g',T,Y,'-b',T,YN,'-
r',T,YK(:,1),'-c'); grid on; % графік вхідного і
вихідного сигналу системи
xlabel('Час, с');ylabel('Амплітуда вихідного
сигналу, Вт');

```

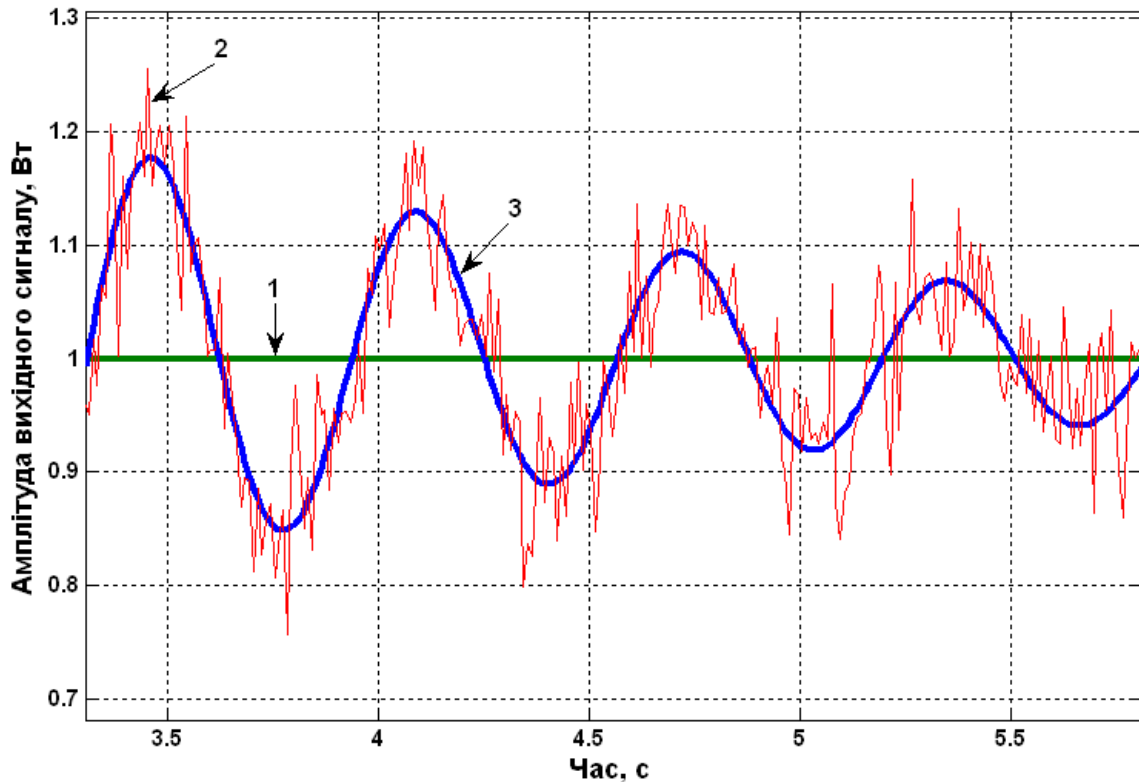


Рис. 7.9.2. Порівняння виміряного і точного значення вихідної координати системи управління: 1- вхідний сигнал (одиничний ступеневий вплив, 1000 відліків), 2- вихід з шумом, 3 - точний вихід системи.

7.9.2. Завдання для лабораторної роботи

1. Згідно свого варіанту обрати структурну схему системи автоматичного управління (додаток 1) та параметри цієї схеми (додаток 2).
2. Зробити розрахунок фільтра Калмана.
3. Виконати моделювання системи управління зі спостерігачем.
4. Побудувати графіки виміряного і точного вихідних сигналів координат ОУ та зашумленого і оціненого значення виходу системи.
5. Порівняти графіки та зробити висновки.

7.9.3. Зміст звіту

1. Назва та мета роботи.
2. Структурна схема системи автоматичного управління згідно свого завдання.
3. Текст програми моделювання.
4. Графіки виміряного і точного вихідних сигналів та зашумленого і точного значення виходу системи координат ОУ.
5. Висновки.

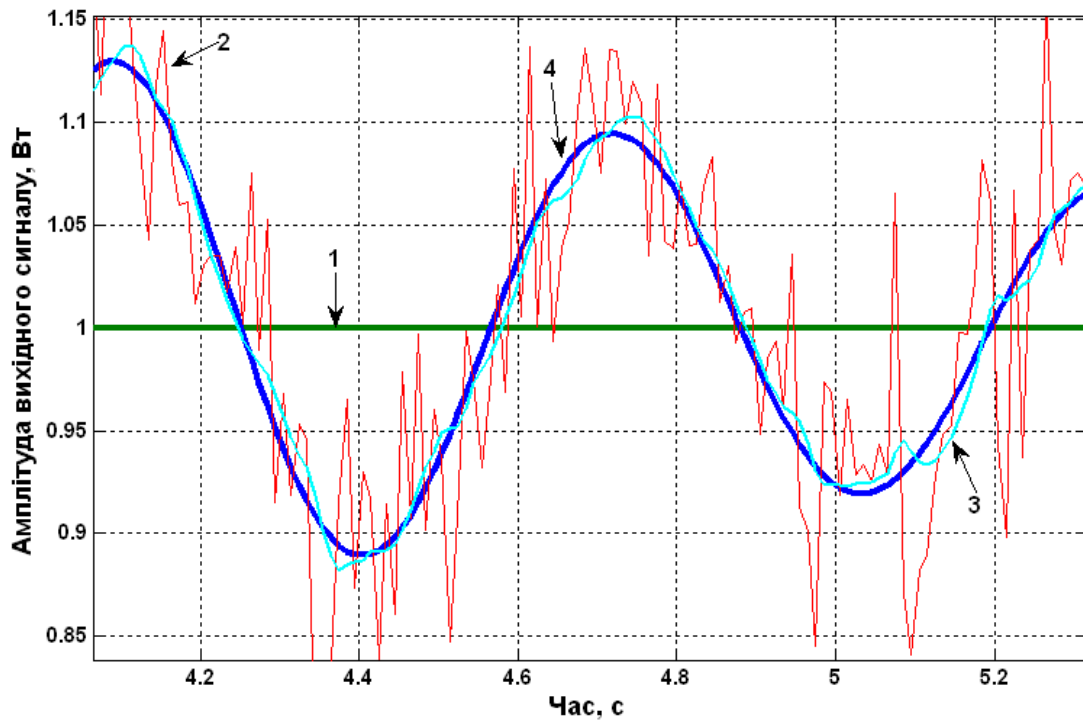


Рис. 7.9.3. Порівняння зашумленого і точного значення виходу системи, виміряного значення і результату оцінювання фільтром Калмана вихідної координати об'єкта системи управління: 1- вхідний сигнал (одиничний ступеневий вплив, 1000 відліків), 2- вихід з шумом, 3 - оцінка виходу системи у фільтрі Калмана, 4 - точний вихід системи.

7.9.4. Контрольні питання

1. Що являє собою фільтр Калмана?
2. Які функції є в пакеті MATLAB для синтезу фільтра Калмана?
3. Які основні блоки входять в структурну схему отримання оцінок координат ОУ?
4. Що таке матриця вимірювання H ?
5. Назвіть матриці об'єкта керування.

7.10. Лабораторна робота 10. Застосування експоненціального згладжування для вимірювання поточних координат об'єктів управління

Мета роботи: ознайомитися з можливостями підвищення точності визначення поточних координат та параметрів руху об'єктів управління

7.10.1. Теоретичні відомості

7.10.1.1. Основні відомості з теорії експоненційного згладжування

Метод експоненційного згладжування – це метод послідовного згладжування параметрів траєкторії руху ОУ за допомогою фільтра с постійною пам'яттю. Принцип роботи цього методу полягає в наступному: попередні значення параметрів враховуються зі спадаючими по експоненційному закону вагами.

Для довільного скалярного параметра θ формула експоненційного згладжування має вид:

$$\hat{\theta}_n = (1 - \xi)\theta_n + \xi f[\hat{\theta}_{n-1}, \hat{\theta}_{n-2}, \dots, \hat{\theta}_{n-s}], \quad (7.10.1)$$

де θ_n – вимірне значення параметра на n -кроці; $\hat{\theta}_n$ - згладжене значення параметра в n -м огляді; $f[\dots]$ - деяка функція попередніх згладжених значень параметрів; ξ – стала величина, що має сенс коефіцієнта згладжування.

У найпростішому вигляді для знаходження наступного загладжуваного значення параметру використовується тільки поточне вимірне значення та попереднє згладжене значення. Тоді, формула (7.10.1) приймає наступний вид:

$$\hat{\theta}_n = (1 - \xi)\theta_n + \xi\hat{\theta}_{n-1}. \quad (7.10.2)$$

Якщо формулу (2) записати у вигляді:

$$\hat{\theta}_n = (1 - \xi)\theta_n + \xi(1 - \xi)\theta_{n-1} + \xi^2(1 - \xi)\theta_{n-2} + \dots + \xi^n(1 - \xi)\theta_0 = \sum_{i=0}^n \eta_i \theta_{n-i}, \quad (7.10.3)$$

де $\eta_i = \xi^i(1 - \xi)$ - вагова функція, що враховує поточне та попередні виміряні значення параметрів руху ОУ.

На рис. 7.10.1 зображені графіки вагової функції η_i при різних значеннях $\xi = 0,9; 0,5; 0,1$. З рисунку бачимо: чим менше ξ , тим в менший ступені враховуються результати попередніх вимірювань (менше згладжування), та навпаки, чим більше ξ , тим більше згладжування, тобто більше результати попередніх вимірювань параметра враховуються в більшій мірі при обчисленні згладженого поточного значення. Тому коефіцієнт ξ отримав назву експоненційного коефіцієнта згладжування.

При згладжування параметру, що змінюється за лінійним законом формула експоненційного згладжування набуває вигляду:

$$\hat{\theta}_n = (1 - \xi)\theta_n + \xi f[\hat{\theta}_{n-1}, \hat{\theta}_{n-2}]. \quad (7.10.4)$$

Операція $f[...]$ над попередніми згладженими значеннями параметру повинна в цьому випадку забезпечити знаходження екстрапольованого значення параметра на момент t_n останнього вимірювання.

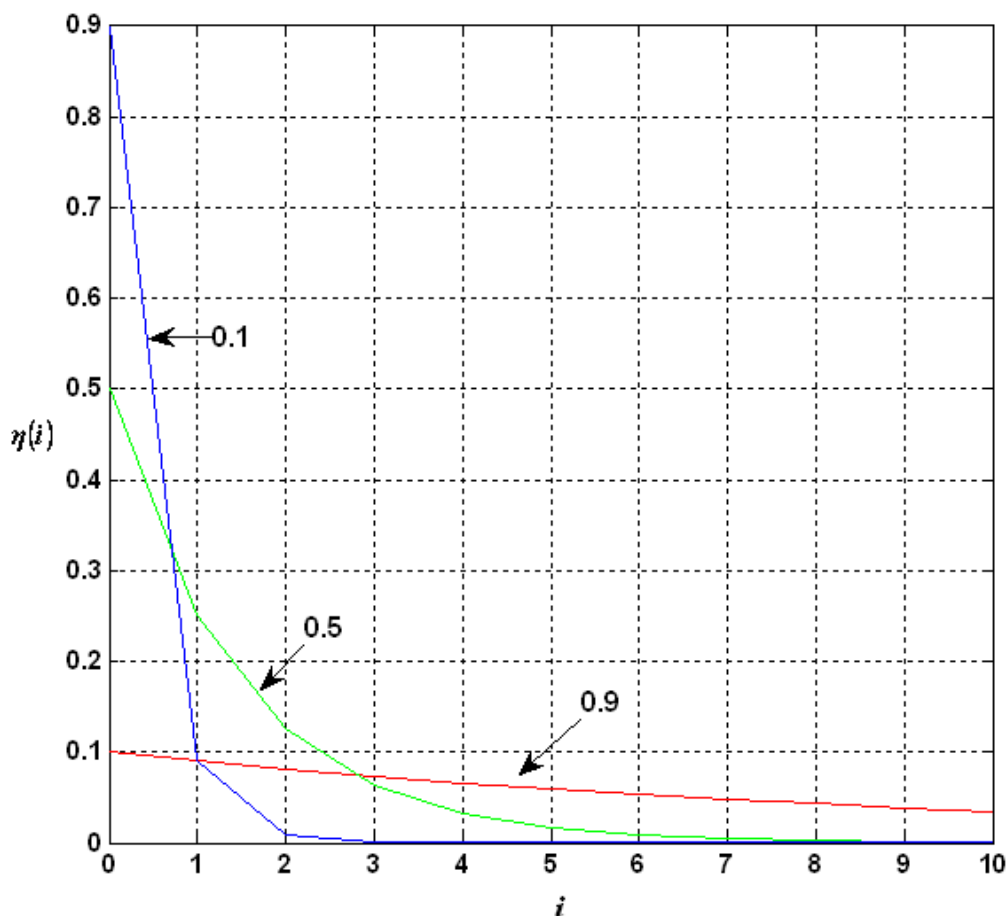


Рис. 7.10.1. Графіки вагової функції η_i при різних значеннях ξ

7.10.1.2. Помилки експоненціального згладжування параметрів руху ОВ. Вибір оптимального значення ξ .

Якість експоненційного згладжування оцінюється за величиною випадкових та динамічних помилок згладжування. Миттєве значення сумарної помилки згладжування параметра запишемо у вигляді

$$\Delta \hat{\theta}_{\Sigma_n} = \hat{\theta}_n - \theta_n^{(0)}, \quad (7.10.5)$$

де $\theta_n^{(0)}$ – істинне значення параметра у момент часу t_n ;

$$\hat{\theta}_n = \sum_{i=0}^n \eta_i \theta_{n-i}, \quad (7.10.6)$$

де θ_{n-i} – результати вимірювання (розрахунку) параметрів, які представляють собою адитивну суміш дійсних значень $\theta_{n-i}^{(0)}$ і помилок вимірювання θ_{n-i} , тобто

$$\theta_{n-i} = \theta_{n-i}^{(0)} + \Delta\theta_{n-i}. \quad (7.10.7)$$

Використовуючи вирази (7.10.6) і (7.10.7), сумарну помилку згладжування параметра можна представити в наступному вигляді:

$$\Delta\theta_{n\Sigma} = \sum_{i=0}^n \eta_i (\theta_{n-i}^{(0)} + \Delta\theta_{n-i}) - \theta_n^{(0)} = \left(\sum_{i=0}^n \eta_i \theta_{n-i}^{(0)} + \Delta\theta_{n-i} \right) + \sum_{i=0}^n \eta_i \theta_{n-i}^{(0)}. \quad (7.10.8)$$

Перший доданок у виразі (7.10.8), що розміщений в дужках, являє собою динамічну складову $\Delta_{\partial}\hat{\theta}_n$ сумарної помилки згладжування, а другий – випадкову складову $\Delta_{\partial}\hat{\theta}_{ec}$ цієї помилки.

Так як випадкова і динамічна складові некорельовані між собою, дисперсія сумарної помилки буде дорівнювати

$$\sigma_{\Sigma}^2 = \sigma_{\hat{\theta}_n}^2 + (\Delta_{\partial}\hat{\theta}_n)^2. \quad (7.10.9)$$

Обчислимо дисперсію випадкової і квадрат динамічної складових похибки визначення параметрів руху ОУ.

Так як математичне сподівання випадкової помилки дорівнює нулю, то дисперсію цієї помилки можна записати у вигляді

$$\sigma_{\hat{\theta}_n}^2 = M \left[\left(\sum_{i=0}^n \eta_i \Delta\theta_{n-1} \right)^2 \right]. \quad (7.10.10)$$

Після елементарних перетворень одержимо

$$\sigma_{\hat{\theta}_n}^2 = \sum_{i=0}^n \sum_{j=0}^n \eta_i \eta_j R_{\theta}(i-j), \quad (7.10.11)$$

де $R_{\theta}(i-j)$ – кореляційний момент випадкових помилок вимірювання параметра.

Припустимо, що згладжуваний параметр обчислюється за двома виміряними значеннями координат, причому кожне вимірне значення бере участь у розрахунку параметра двічі. Тоді, за умови відсутності кореляції між похибками вимірювання, обчислені значення параметра будуть корельовані на двох сусідніх кроках. Безпосереднім підрахунком можна показати, що коефіцієнт кореляції в цьому випадку від'ємний та дорівнює 0,5.

Отже

$$R_{\theta}(i-j) = \begin{cases} \sigma_{\theta}^2 & \text{при } i-j=0, \\ -0,5\sigma_{\theta}^2 & \text{при } |i-j|=1, \\ 0 & \text{при } |i-j|>0, \end{cases} \quad (7.10.12)$$

де σ_{θ}^2 – дисперсія розрахунку параметра по двом виміряним значенням координат.

З урахуванням (12) вираз для дисперсії випадкової помилки можна записати у вигляді

$$\sigma_{\hat{\theta}_n}^2 = \sigma_{\theta}^2 \left[\sum_{i=0}^n \eta_i^2 - \sum_{i=0}^n \eta_i \eta_{i+1} \right]. \quad (7.10.13)$$

Маючи на увазі, що при $n \rightarrow \infty$ (сталій режим роботи)

$$\sum_{i=0}^{\infty} \eta_i^2 = \sum_{i=0}^{\infty} [(1-\xi)\xi^i]^2 = \sum_{i=0}^{\infty} (1-\xi)^2 \xi^{2i} = (1-\xi)^2 (1 + \xi^2 + \xi^4 + \dots), \quad (7.10.14)$$

$$\sum_{i=0}^{\infty} \eta_i \eta_{i+1} = \sum_{i=0}^{\infty} (1-\xi)^2 \xi^{2i+1} = (1-\xi)^2 (\xi + \xi^3 + \xi^5 \dots), \quad (7.10.15)$$

остаточно отримаємо

$$\sigma_{\hat{\theta}_n}^2 = \sigma_{\theta}^2 \frac{(1-\xi)^2}{1+\xi}. \quad (7.10.16)$$

Формула (7.10.16) показує, що дисперсія випадкової помилки експоненціального згладжування параметра тим менше, чим більше коефіцієнт згладжування ξ .

Обчислимо тепер квадрат динамічної помилки згладжування параметра для випадку усталеного руху ОУ за цим параметром.

Коефіцієнти динамічної помилки визначаються з виразу

$$C_i = \frac{(-1)^i}{i!} \mu_i - a_i, \quad (7.10.17)$$

де

$$\mu_i = \sum_{j=0}^{\infty} j^j \eta_j \quad (i=0,1,2,\dots,l), \quad (7.10.18)$$

l – ступінь полінома, що описує функцію $\theta^{(0)}(t)$.

Коефіцієнти a_i мають значення $a_0 = 1, a_1 = p, \dots, a_i = p^i / i!$, де p - число періодів екстраполяції вхідного сигналу; при згладжуванні $p=0, a_0 = 1$.

Для випадку лінійної зміни параметра руху ($j=1$), обчислення коефіцієнтів динамічної помилки дають

$$\mu_0 = \sum_{j=0}^{\infty} j^0 \eta_j = (1-\xi) \sum_{j=1}^{\infty} \xi^j = 1, \quad (7.10.19)$$

тому $C_0 = 0$;

$$\mu_1 = \sum_{j=0}^{\infty} j \eta_j = (1-\xi) \sum_{j=0}^{\infty} j \xi^j = \frac{\xi}{1-\xi}, \quad (7.10.20)$$

тому $C_1 = \frac{\xi}{1-\xi}$.

Таким чином,

$$\Delta_{\partial} \hat{\theta}_n = \frac{\xi}{1-\xi} \Delta^{(1)} \theta_n, \quad (7.10.21)$$

а квадрат цієї помилки визначається за формулою

$$(\Delta_{\partial} \hat{\theta}_n)^2 = \frac{\xi^2}{(1-\xi)^2} (\Delta^{(1)} \theta_n)^2. \quad (7.10.22)$$

З урахуванням виразів (7.10.16) і (7.10.22) дисперсія відносної сумарної помилки згладжувань дорівнює

$$\frac{\sigma_{\Sigma}^2}{\sigma_{\theta}^2} = \frac{(1+\xi)^2}{(1-\xi)} + \frac{\xi^2}{(1-\xi)^2} s^2, \quad (7.10.23)$$

де $s^2 = \frac{(\Delta^{(1)} \theta_n)^2}{\sigma_{\theta}^2}$ - відносна інтенсивність можливих змін параметрів руху.

На рис. 7.10.2 побудовані графіки залежності дисперсії відносної сумарної помилки згладжування параметра від коефіцієнта згладжування ξ при різних інтенсивностях змін цього параметру.

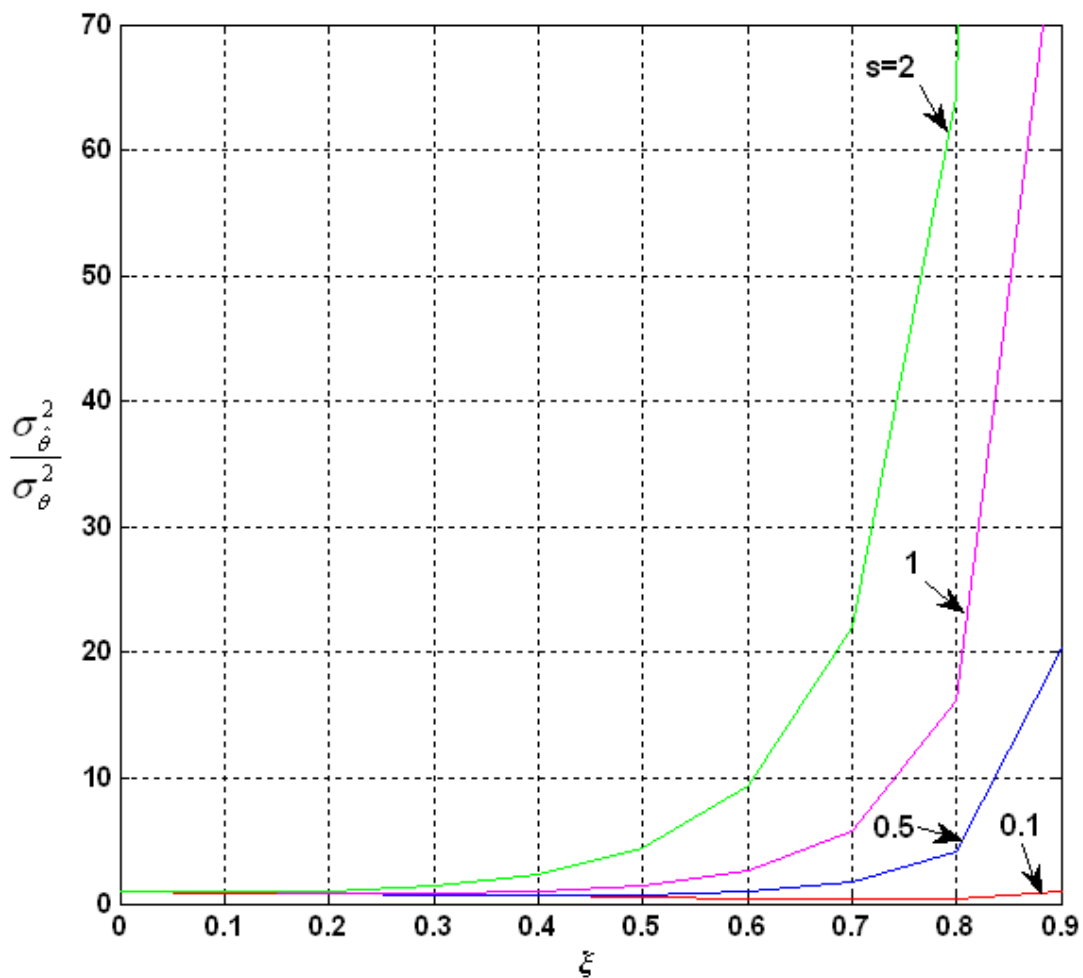


Рис. 7.10.2. Графіки залежності дисперсії відносної сумарної помилки згладжування параметра від коефіцієнта згладжування ξ при різних інтенсивностях маневру

Графіки мають екстремуми типу мінімуму, які зміщуються вліво при збільшенні інтенсивності змін. Звідси можна зробити висновок про доцільність зміни коефіцієнта згладжування ξ з метою підтримки похибки на мінімальному або близькому до мінімального рівня для кожного конкретного значення інтенсивності змін параметра руху ОУ. З графіків похибки також випливає, що вибір коефіцієнтів ξ для кожного значення інтенсивності змін не критичний, так як екстремуми графіків пологі.

Для знаходження оптимального значення ξ_{opt} за критерієм мінімуму дисперсії сумарної помилки продиференціюємо вираз (7.10.23) по ξ і прирівняємо результат до нуля. Остаточне рівняння для ξ_{opt} має вигляд

$$(1 - \xi_{opt}^2)(1 - \xi_{opt})^3 + 2(1 - \xi_{opt})^4 - 2\xi_{opt}(1 + \xi_{opt})^2 s^2 = 0. \quad (7.10.24)$$

Використовуючи це рівняння, можна вибирати оптимальне значення коефіцієнта згладжування ξ_{opt} , якщо відома інтенсивність змін параметра руху ОУ.

7.10.1.3. Рівняння згладжування в дискретній формі

Призначення цифрового обчислювального пристрою, що використовується як елемент системи регулювання або управління, полягає в обчисленні згладжених значень дискретних даних, які містять завади.

Позначимо через θ_n значення дискретної величини, виміряної в рівновіддалені один від одного інтервали часу, а через $\hat{\theta}_n$ – згладжені значення тієї же величини. Процедура згладжування буде полягати в екстраполяції або передбаченні дискретної величини θ_n в даний момент часу n по значенню незгладженої величини θ_n в цей момент часу та по попередніх згладжених значеннях $\hat{\theta}_{n-v}$, тобто

$$\hat{\theta}_n = (1 - \xi)\theta_n + \xi f(\theta_{n-1}, \theta_{n-1}, \dots, \theta_{n-s-1}), \quad (7.10.24)$$

де $\xi \leq 1$ – стала величина, коефіцієнт згладжування. Початкові значення $\hat{\theta}_n$ при $n=0, 1, \dots, s$ можуть бути обрані довільно та, зокрема, вони можуть дорівнювати незгладженим значенням $\hat{\theta}_n$. З (7.10.24) випливає, що процес згладжування описується різницеvim рівнянням.

Для лінійного передбачення функція $f[...]$ являє собою лінійну функцію своїх аргументів. Щоб визначити конкретний вид цієї функції, припустимо, що передбачення базується на апроксимації $\hat{\theta}_n$ поліномом s -го ступеня. Позначимо цей поліном через $\varphi(\bar{t})$. Для дискретних значень

$$\varphi_m = \hat{\theta}_m, \quad m = n-1, n-2, \dots, n-s-1. \quad (17.0.25)$$

Функція $f[...]$ повинна співпадати з дискретними значеннями цього полінома при $\bar{t} = n$, тобто

$$f(\theta_{n-1}, \theta_{n-2}, \dots, \theta_{n-s-1}) = \varphi_n. \quad (7.10.26)$$

Але для будь-якого полінома ступені s різниця $s+1$ -го порядку тотожно дорівнює нулю при будь-якому m , тобто

$$\Delta^{s+1} \varphi_m = 0. \quad (7.10.27)$$

Значення решітчастої функції отримуємо з (7.10.27) через значення її ординат, при $m=n-s-1$:

$$\Delta^{s+1} \varphi_{n-s-1} = \sum_{v=0}^{s+1} (-1)^v \binom{s+1}{v} \varphi_{n-v} = 0, \quad (7.10.28)$$

Запишемо тотожність:

$$\varphi_n = \varphi_n - \Delta^{s+1} \varphi_{n-s-1} = - \sum_{v=0}^{s+1} (-1)^v \binom{s+1}{v} \varphi_{n-v}. \quad (7.10.29)$$

Підставляючи значення φ_n з (7.10.29) в (7.10.26) та враховуючи (7.10.25), отримуємо вираз для $f[\dots]$:

$$f(\theta_{n-1}, \theta_{n-2}, \dots, \theta_{n-s-1}) = \sum_{v=0}^{s+1} (-1)^v \binom{s+1}{v} \hat{\theta}_{n-v}. \quad (7.10.30)$$

Після визначення функції $f[\dots]$ рівняння (7.10.24) можна представити у вигляді

$$\hat{\theta}_n = (1 - \xi) \theta_n - \xi \sum_{v=0}^{s+1} (-1)^v \binom{s+1}{v} \hat{\theta}_{n-v}. \quad (7.10.31)$$

7.10.1.4. Еквівалентна цифрова система для оцінки параметрів руху ОУ

Використовуючи теорему зсуву для рівняння (7.10.31), знайдемо рівняння згладжування відносно зображень параметра руху ОУ:

$$Z^*(q) = (1 - \xi) F^*(q) - \xi \left(\sum_{v=0}^{s+1} (-1)^v \binom{s+1}{v} e^{-qv} \right) Z^*(q). \quad (7.10.32)$$

де $Z^*(q)$ та $F^*(q)$ – зображення решітчастих функцій $\hat{\theta}_n$ та θ_n .

Припустимо, що $\theta_{-n} = 0$. Введемо позначення

$$K^*(q, 0) = \xi \sum_{v=0}^{s+1} (-1)^v \binom{s+1}{v} e^{-qv} = \xi \left\{ (1 - e^{-q})^{s+1} - 1 \right\}. \quad (7.10.33)$$

Тоді рівняння (7.10.32) можна остаточно привести до вигляду:

$$Z^*(q) = \frac{1 - \xi}{1 + K^*(q, 0)} F^*(q). \quad (7.10.34)$$

Це рівняння відповідає деякій амплітудній імпульсній системі, що еквівалентна цифровому обчислювальному пристрою або цифровому фільтру, що здійснює вказаний процес згладжування (рис. 7.10.3). $K^*(q, 0)$ можна розглядати як передаточну функцію розімкненої амплітудної імпульсної системи.

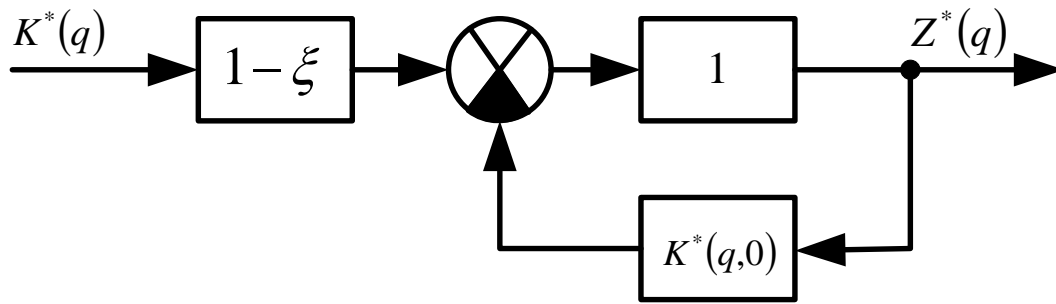


Рис. 7.10.3. Схема згладжування з використанням цифрового обчислювального пристрою

Передаточна функція замкнутої цифрової системи, згідно (7.10.34), дорівнює:

$$K_e^*(q, 0) = K_u^*(q, 0) = \frac{1 - \xi}{1 + K^*(q, 0)} = \frac{1 - \xi}{1 + \xi[(1 - e^{-q})^{s+1} - 1]}, \quad (7.10.35)$$

що відповідає виконанню програми у цифровому обчислювальному пристрої, який виконує процес згладжування, описаний вище.

7.10.1.5. Оцінка ефективності згладжування

Важливою оцінкою якості згладжування може слугувати відношення середніх значень квадратів згладженої та незгладженої величини в тому випадку, коли незгладжені дані представляють дискретний випадковий процес з постійною спектральною щільністю. Чим менше це відношення в порівнянні з одиницею, тим ефективніший процес згладжування; якщо це відношення більше одиниці, то згладжування не відбувається.

Якщо незгладжені величини θ_n представляють собою дискретний випадковий процес з постійною спектральною щільністю $S_{ff}^*(\bar{\omega}) = \sigma^2$ та з середнім значенням, що дорівнює нулю, то середнє значення квадрата цієї помилки дорівнює $\bar{\theta}_n^2 = \sigma^2$.

Середнє значення квадрата відхилення згладженої величини може бути знайдено по формулі

$$\bar{\theta}_n^2 = \frac{1}{\pi} \int_0^\pi |K_u^*(j\bar{\omega}, 0)|^2 S_{ff}^*(\bar{\omega}) d\bar{\omega}. \quad (7.10.36)$$

або в розгорнутому вигляді

$$\bar{\theta}_n^2 = \frac{1}{\pi} \int_0^\pi \left| \frac{1 - \xi}{1 + \xi[(1 - e^{-j\bar{\omega}})^{s+1} - 1]} \right|^2 \sigma^2 d\bar{\omega}. \quad (7.10.37)$$

Для розрахунку цього значення скористаємося формулою:

$$\begin{aligned}
P^*(q) &= K_y^*(q,0)\sigma = \frac{1-\xi}{1+\xi \sum_{v=1}^{s+1} (-1)^v \binom{s+1}{v} e^{-qv}} \sigma = \\
&= \frac{(1-\xi)e^{(s+1)q}}{e^{(s+1)q} + \xi \sum_{v=1}^{s+1} (-1)^v \binom{s+1}{v} e^{(s+1-v)q}}.
\end{aligned}
\tag{7.10.38}$$

Використовуючи формулу для сумарної оцінки та вважаючи, що в ній $l = s+1$, отримуємо:

$$b_k = 1 - \xi \text{ при } k = s+1,$$

$$b_k \equiv 0 \text{ при } k < s+1$$

та $a_{s+1} = 1$, $a_{s+1-v} = (-1)^v \binom{s+1}{v}$, де $(v=1, 2, \dots, s+1)$. В результаті, отримуємо після обчислення для різних s :

$$\left. \begin{aligned}
s=0: \quad \frac{\overline{\hat{\theta}_n^2}}{\sigma^2} &= \frac{1-\xi}{1+\xi}; \\
s=1: \quad \frac{\overline{\hat{\theta}_n^2}}{\sigma^2} &= \frac{1+\xi}{1+3\xi}; \\
s=2: \quad \frac{\overline{\hat{\theta}_n^2}}{\sigma^2} &= \frac{(1-\xi)(1+4\xi)}{(1-2\xi)(1+7\xi)}; \\
s=3: \quad \frac{\overline{\hat{\theta}_n^2}}{\sigma^2} &= \frac{1+8\xi-25\xi^2}{(1-5\xi)(1+15\xi)}.
\end{aligned} \right\}
\tag{7.10.39}$$

В цих формулах ξ не повинно перевищувати граничного значення.

Залежності $\frac{\overline{\hat{\theta}_n^2}}{\sigma^2}$ від ξ приведені на (рис. 7.10.4).

Ці залежності дозволяють при заданому s обрати те значення ξ , яке забезпечує задану точність визначення поточної координати та параметрів руху ОУ.

З рис. 7.10.4 випливає, враховуючи умови стійкості, що $\frac{\overline{\hat{\theta}_n^2}}{\sigma^2}$ може бути менше одиниці тільки при $s=0, 1, 2$ та, означає, що тільки при цих значеннях s буде здійснювати згладжування.

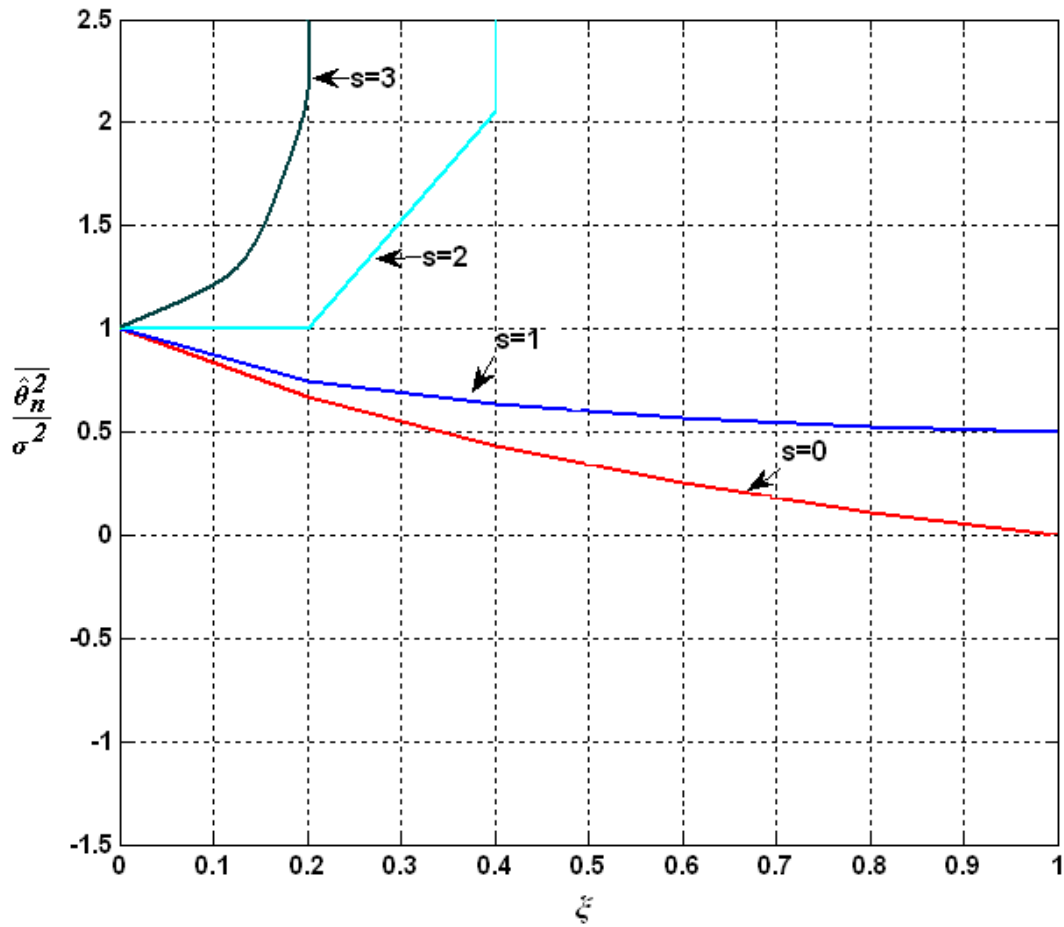


Рис. 7.10.4. Залежність відношень середніх значень квадратів нормованої похибки при різних значеннях s

При $s \geq 3$ описаний спосіб не може забезпечити згладжування, так як незалежно від величини ξ завжди $\frac{\overline{\hat{\theta}_n^2}}{\sigma^2}$ більше одиниці. Цей недолік може бути усунений, якщо проводити передбачення або екстраполяцію не по одному значенню незгладженої величини θ_n , а по декільком $\theta_n, \theta_{n-1}, \theta_{n-2}$, тобто, якщо замінити закон згладжування (7.10.31) на (7.10.40)

$$\hat{\theta}_n = (1 - \xi) \sum_{v=0}^{s-1} \zeta_v \theta_{n-v} - \xi \sum_{v=1}^{s+1} (-1)^v \binom{s+1}{v} \hat{\theta}_{n-v}, \quad (7.10.40)$$

де стійкість процесу згладжування не залежить від коефіцієнтів ζ_v .

7.10.1.6. Реалізація експоненційного згладжування в програмі MATLAB

Для реалізація експоненційного згладжування в програмі MATLAB не має спеціальних функцій, тому це необхідно здійснювати за допомогою програми, що розроблена на основі формул експоненційного згладжування (додаток 3).

Графіки функцій, що отримані після моделювання експоненціального згладжування в програмі MATLAB, представлені на рис. 7.10.5-7.10.7 .

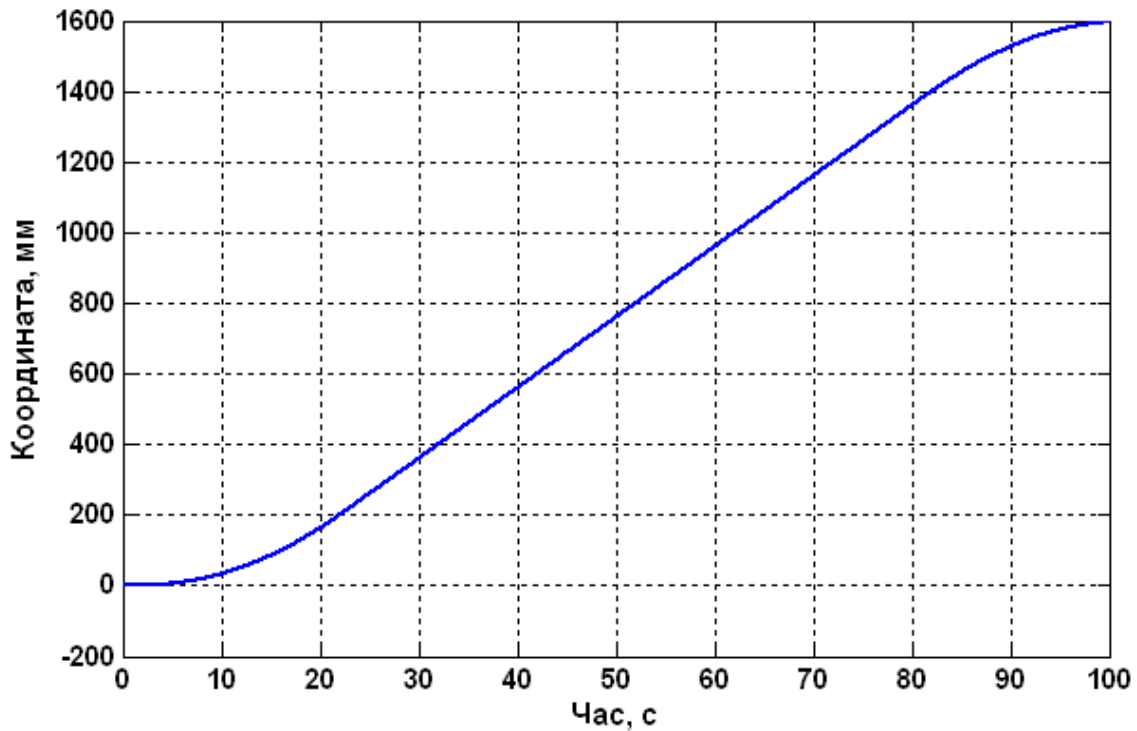


Рис. 7.10.5. Графік зміни координати ОУ в часі

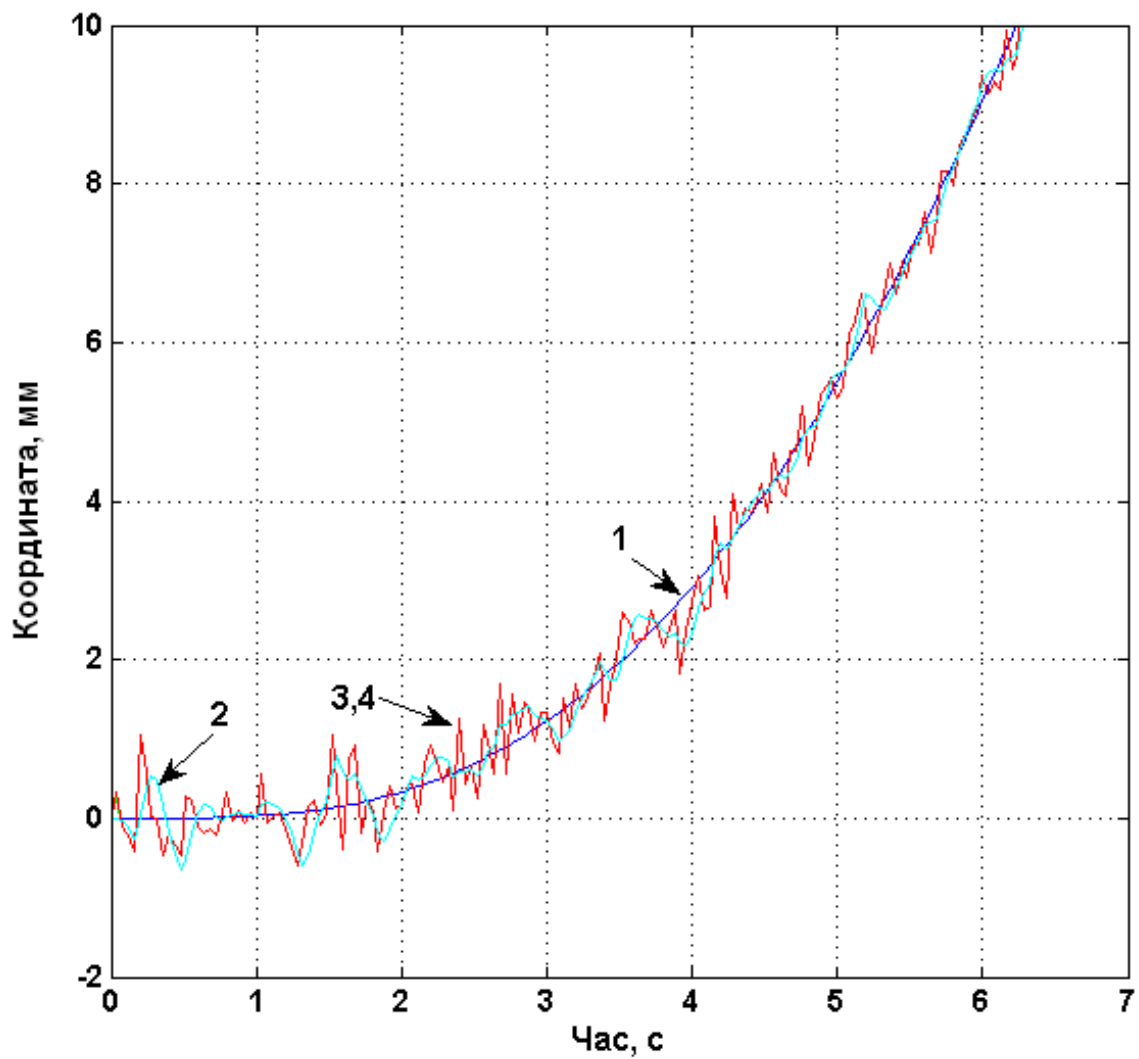


Рис. 7.10.6. Графіки: 1- точного значення координати ОУ , 2- оцінка на основі експоненціального згладжування, 3 – оцінка поточних значень координат фільтром Калмана, 4 - результат вимірювань координат.

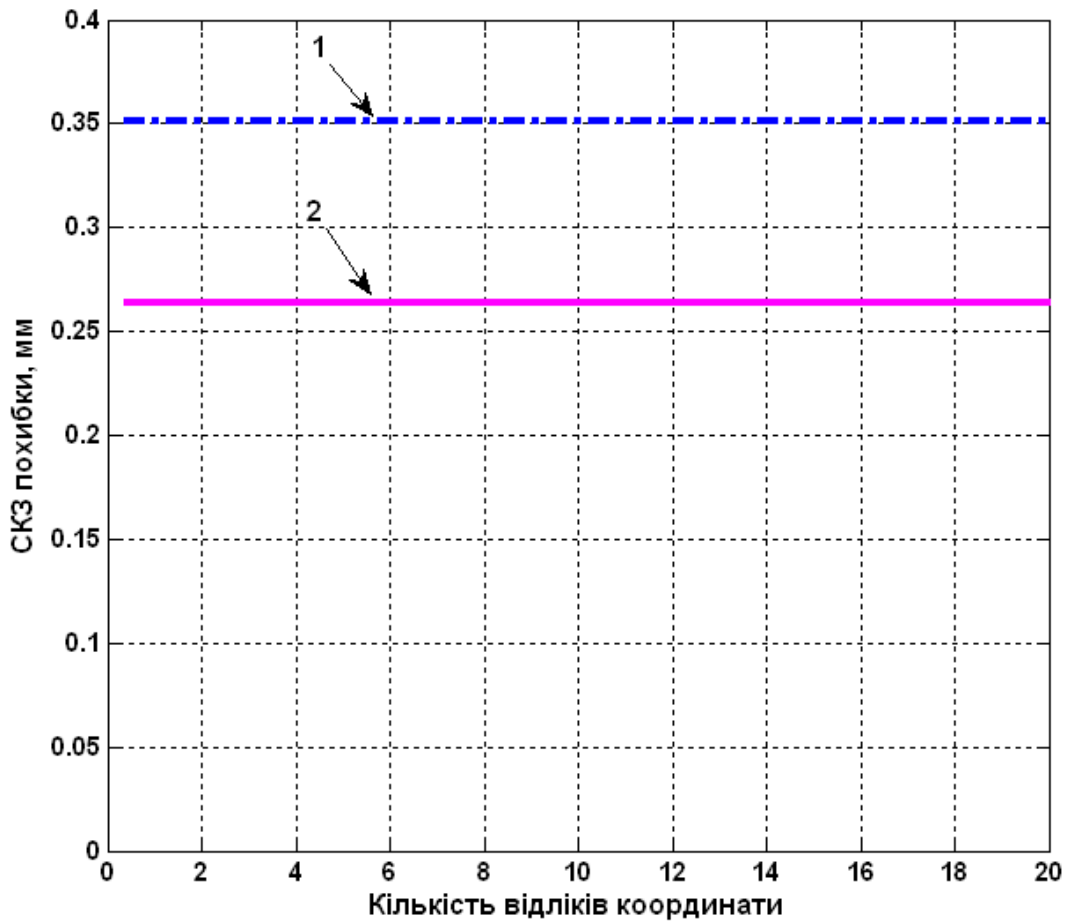


Рис. 7.10.7. СКЗ похибки оцінки: 1 – оцінка фільтра Калмана, 2 – оцінка експоненціального згладжування.

7.10.2. Завдання для лабораторної роботи

1. Згідно свого варіанту початкових даних системи автоматичного управління (Таблиця 7.10.1) виконати моделювання системи управління з точним входом, з шумом, з експоненційним згладжуванням та з фільтром Калмана (додаток В).

2. Побудувати графіки точного вихідного сигналу координат ОУ та з шумом, оціненого значення фільтром Калмана та експоненціальним згладжуванням, а також графіки СКЗ похибки оцінок для фільтра Калмана та для експоненційного згладжування.

3. Порівняти графіки та зробити висновки.

Таблиця 7.10.1

№ варіанту	Початкова координата x_0 , мм	Швидкість руху ОУ v_0 , мм/с	Кількість попередніх відліків в формулі експоненційного згладжування
1	0	10	1
2	35	11	2
3	70	12	1
4	105	14	2
5	140	15	1
6	175	16	2
7	210	17	1
8	245	18	2
9	280	19	1
10	315	20	2
11	350	21	1
12	385	22	2
13	420	23	1
14	455	24	2
15	490	25	1
16	525	26	2
17	560	27	1
18	595	28	2
19	630	29	1
20	665	30	2
21	700	31	1
22	735	32	2
23	770	33	1
24	805	34	2
25	840	35	1
26	875	36	2
27	910	37	1
28	945	38	2
29	980	39	1
30	1000	40	2

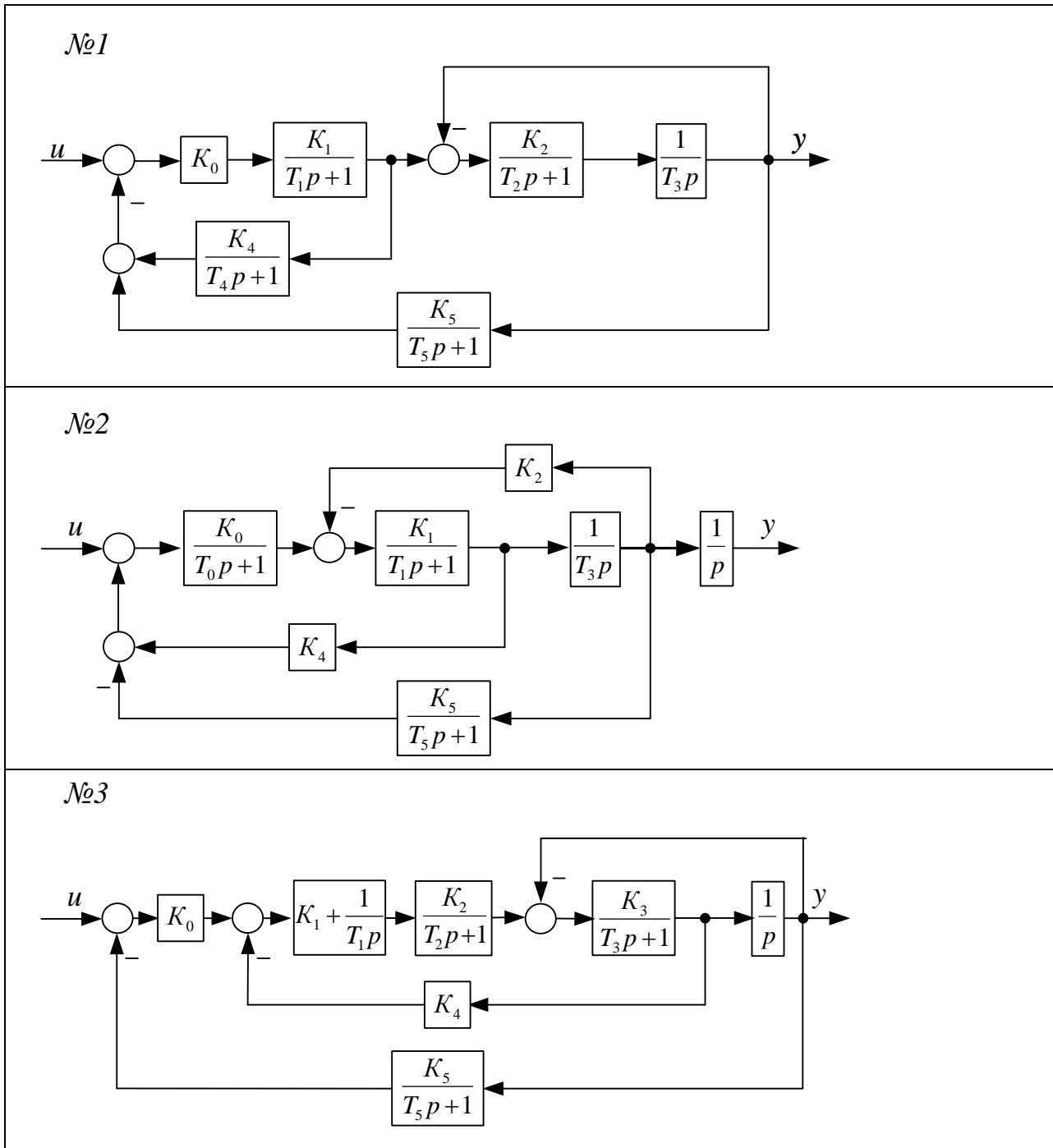
7.10.3. Зміст звіту

1. Назва та мета роботи.
2. Початкові дані параметрів руху ОУ згідно свого завдання.
3. Текст програми моделювання.
4. Графіки точного вихідного сигналу координат ОУ та сигналу з шумом, оціненого значення фільтром Калмана та експоненціальним згладжуванням. Графіки СКЗ похибки оцінок у фільтру Калмана та у експоненціальному згладжуванні.
5. Висновки.

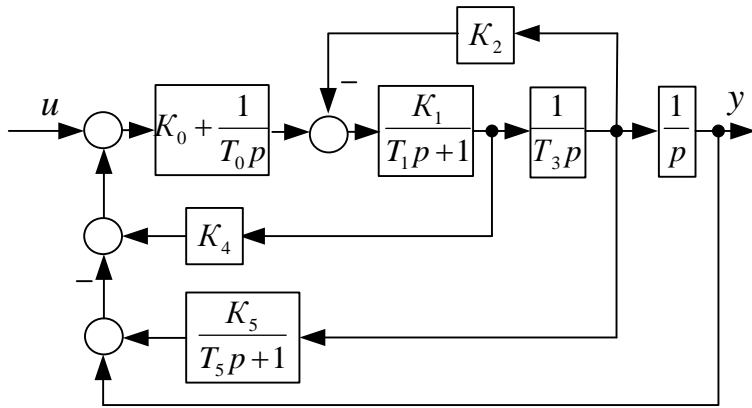
7.10.4. Контрольні питання

1. Що являє собою експоненційне згладжування?
2. Які реалізувати моделювання експоненційного згладжування в програмі MATLAB?
3. Які значення може приймати коефіцієнт згладжування?
4. Як змінюються графіки експоненційного згладжування при різних значеннях інтенсивності змін параметрів руху ОУ?
5. Які значення може приймати інтенсивність змін параметрів руху?

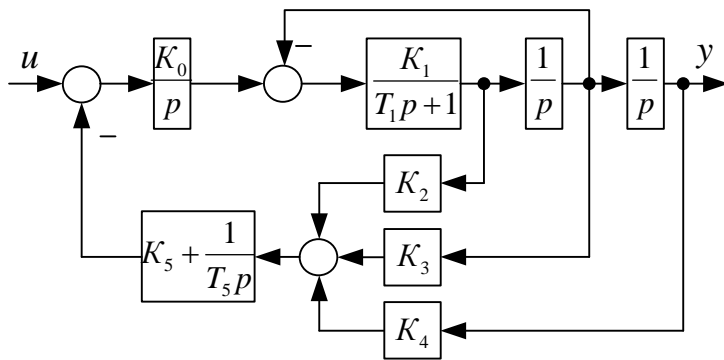
Структурні схеми систем автоматичного управління



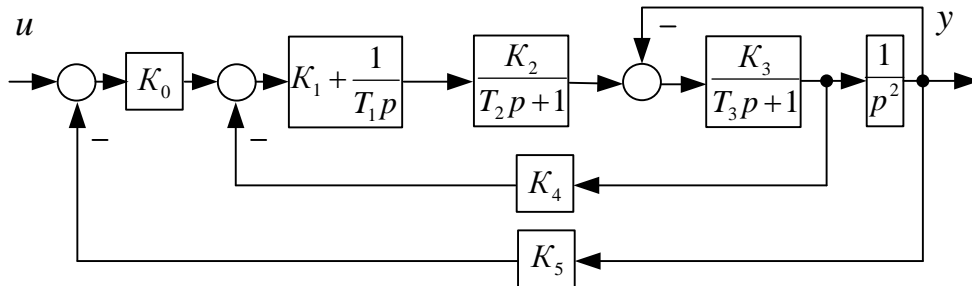
№4



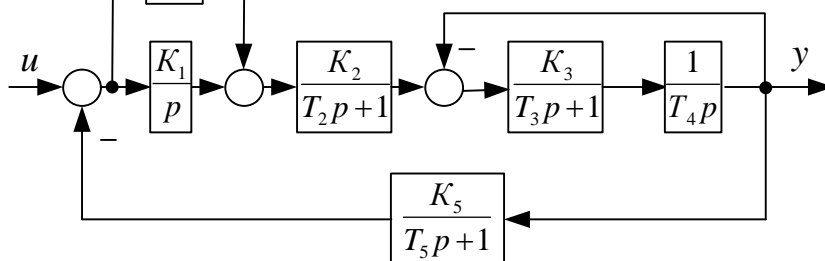
№5



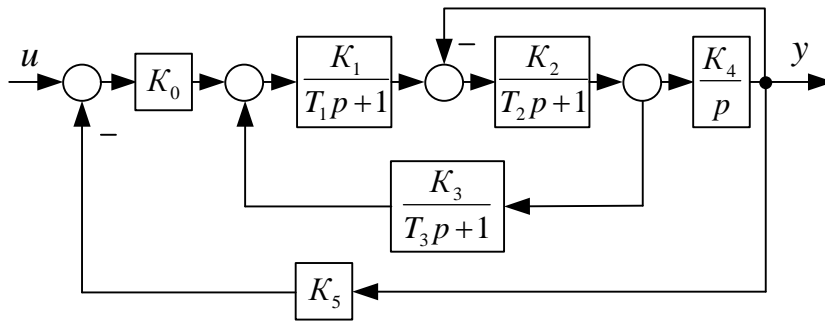
№6



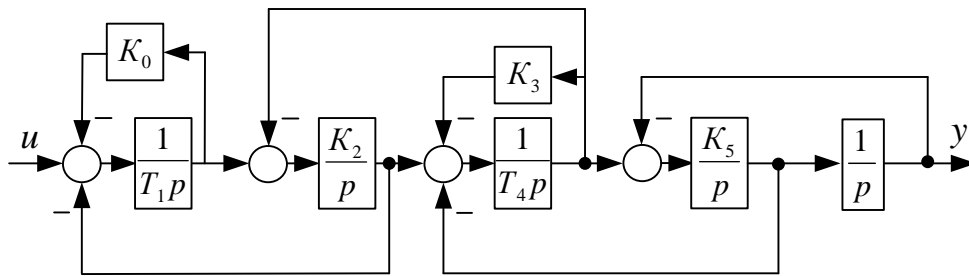
№7



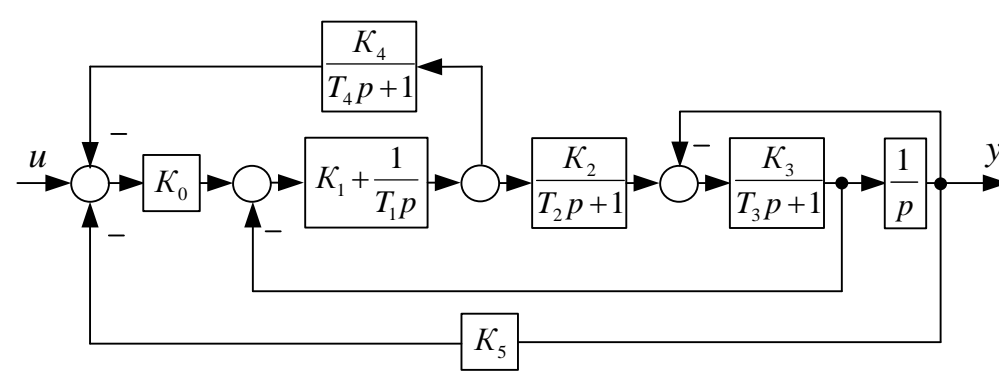
№8



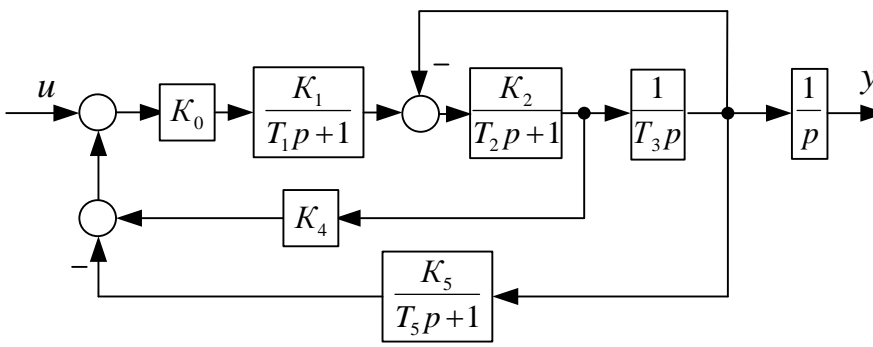
№9



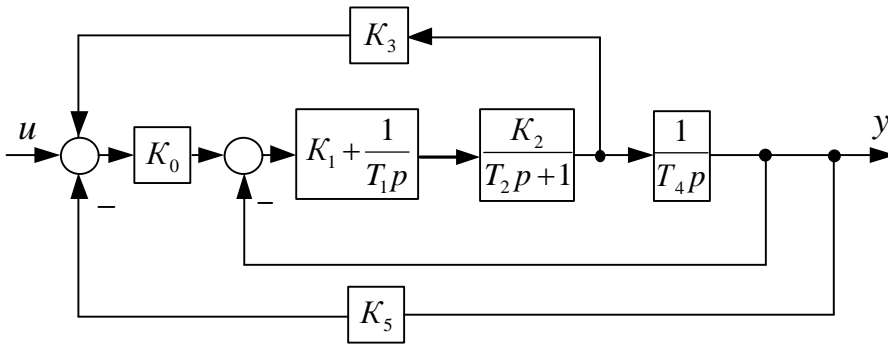
№10



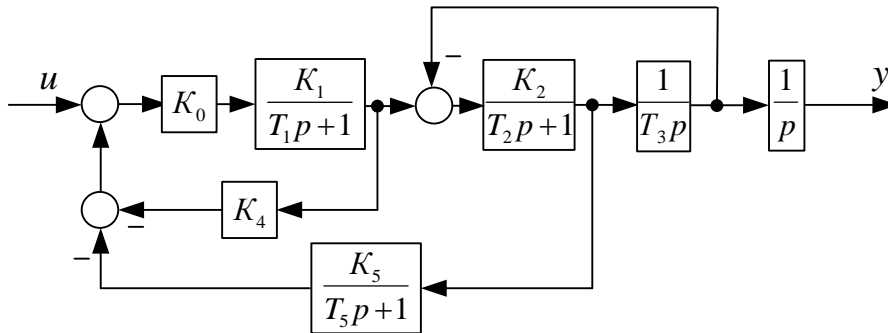
№11



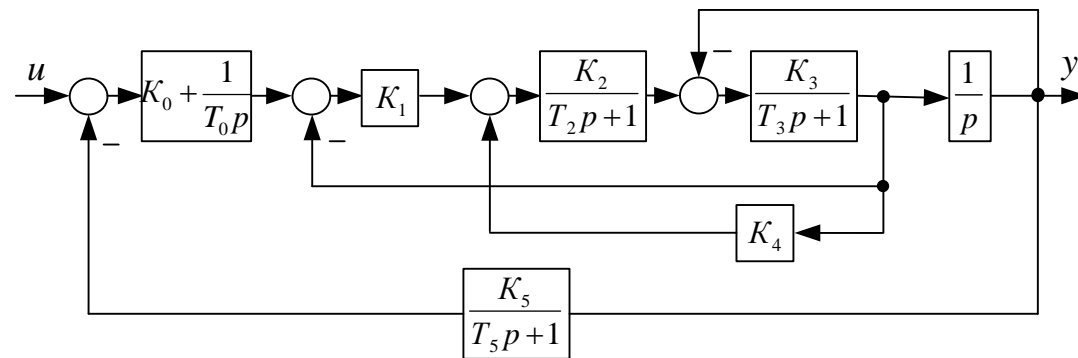
№12



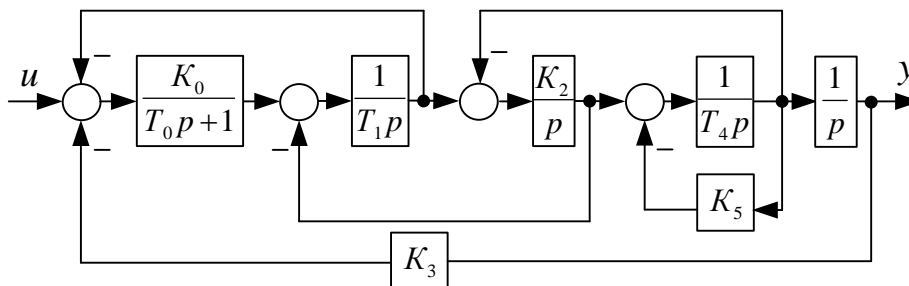
№13



№14



№15



Параметри структурних схем систем автоматичного управління

Варіант	№ схеми	Параметри ланок структурної схеми											
		K ₀	T ₀	K ₁	T ₁	K ₂	T ₂	K ₃	T ₃	K ₄	T ₄	K ₅	T ₅
1	15	7,3	0,2	-	0,2	4,0	-	0,5	-	-	0,2	0,7	-
2	13	5,0	-	1,0	0,2	4,1	0,8	-	0,3	0,2	-	0,1	0,2
3	11	0,4	-	0,4	1,0	2,2	0,6	-	2,3	0,1	-	2,0	0,1
4	9	0,5	-	-	0,4	2,5	-	1,5	-	-	0,5	5,7	-
5	7	1,4	-	1,0	-	8,7	0,1	0,7	0,2	-	2,5	0,5	0,1
6	5	0,3	-	7,7	0,3	3,6	-	1,5	-	8,1	-	9,0	0,8
7	3	4,8	-	2,5	0,1	0,5	0,2	0,6	0,1	1,5	-	1,5	0,2
8	1	10	-	7,1	0,5	5,0	0,1	-	2,0	0,5	0,2	0,3	0,1
9	2	2,1	0,5	1,5	0,3	0,8	-	1,0	1,0	0,1	-	0,5	0,1
10	4	3,8	0,2	7,7	1,3	70	-	-	0,4	1,2	-	2,0	0,3
11	6	0,1	-	9,5	0,2	2,5	0,5	0,8	0,3	0,1	-	0,3	-
12	8	1,2	0,1	9,5	0,5	1,0	0,1	0,5	0,1	2	-	0,1	-
13	10	2,7	-	3,5	1,0	2,7	1,3	4,8	0,7	0,3	1,5	0,5	-
14	12	5,2	-	0,7	1,0	5,3	0,7	0,1	-	-	2,5	0,1	-
15	14	1,1	3,0	0,9	-	3,5	0,2	1,2	0,7	0,1	-	3,6	0,1
16	15	2,1	0,3	-	0,6	8,5	-	1,7	-	-	0,4	0,1	-
17	13	2,7	-	1,5	1,7	8,3	2,9	-	0,7	0,5	-	0,2	0,1
18	11	0,7	-	0,3	0,4	1,6	1,2	-	0,4	0,2	-	2,6	0,1
19	9	0,2	-	-	0,6	7,5	-	0,6	-	-	0,8	0,9	-
20	7	7,5	-	1,0	-	3,2	0,2	9,5	0,1	-	0,5	8,2	0,2
21	5	0,1	-	2,1	0,5	5,1	-	7,5	-	3,9	-	5,0	0,6
22	3	1,8	-	8,5	0,5	1,5	0,3	0,5	0,5	5,0	-	9,5	0,1
23	1	2,1	-	5,0	0,2	10	0,3	-	0,5	3,0	0,8	0,1	0,5
24	2	0,5	0,8	0,7	0,1	1,5	-	1,0	0,2	0,5	-	8,5	0,1
25	4	12	0,1	0,1	0,5	100	-	-	0,5	0,1	-	4,6	0,1
26	6	0,2	-	7,5	0,1	2,5	0,1	5,0	0,7	0,2	-	0,2	-
27	8	6,9	0,3	4,1	0,7	7,3	0,2	3,8	0,5	3,3	-	0,2	-
28	10	2,5	-	2,9	1,0	9,5	1,4	9,5	0,4	0,1	0,2	1,0	-
29	12	0,2	-	0,2	1,0	5,3	0,2	0,2	-	-	1,1	0,1	-
30	14	2,8	2,0	1,8	-	7,5	0,6	7,2	0,2	0,2	-	3,6	0,1

**Програма експоненціального згладжування та оцінювання значень
поточних координат та параметрів руху об'єктів**

```

% Початкові дані
Nos=[9 25 51 101 151 201 301 401 501]';
Type_V=3;          % рух ОБ 1 - з постійною швидкістю 2 -
стрибокподібна зміна швидкості від 0 до максимального значення
                    % 3 - переміщення на задану відстань (для 3
час спостереження 200 с)
FK_P=4;           % порядок матриць для фільтра Калмана min 3
EKS_P=2;         % кількість попередніх значення для
експоненціального згладжування
ksi=0.7;         % коефіцієнт згладжування
jjj=4;          % виведення графіків параметрів руху для Nos=jjj
FKMean=100;     % усереднення відліків для оцінки точності ФК
v=20;          % максимальна швидкість, мм/с
a_max=1;        % максимальна швидкість, мм/с^2
x0=0;          % початкова координата ОБ ФК
v0=v;          % початкова швидкість ОБ ФК
a0=a_max;      % початкове прискорення ОБ ФК
PARAM=[x0 v0 a0 0 0 0];
dt1=0.04;      % крок дискретності
t1=(0:dt1:100)'; % час спостереження
SigmaNoise=0.6/sqrt(3); % СКЗ шуму вимірювань координати
Sigma_Dist_x=0%3.33; % СКЗ збурень координати 5%
Sigma_Dist_v=0%1.67; % СКЗ збурень швидкості 5%
Sigma_Dist_a=0; % СКЗ збурень прискорення
Sigma_Dist_x0=0%0.1; % СКЗ початкових умов координати ФК
Sigma_Dist_v0=0%0.1; % СКЗ початкових умов швидкості ФК
Sigma_Dist_a0=0%0.0001; % СКЗ початкових умов прискорення ФК
TWx=0.12; TWv=0.12; % постійні часу (секунд) аперіодичної
ланки для фільтрації збурень - корельовані збурення
Tu=1; % постійні часу (секунд) аперіодичної ланки 2-го
порядку для фільтрації управляючого сигналу - врахування
інерційності ОБ
randn('state',0);

% Математична модель руху неперервного ОБ 2-го порядку -
неперервна система 2-го порядку
% в просторі стану. Вхід - швидкість і прискорення. Збурення по
входу - для швидкості і прискорення
% Моделюються рівноприскорений рух (U1=0 U2=a(t)) і рівномірний
рух (U1=v(t) U2=0)
% dX/dt=AX+BU+GW
% Y= CX+DU+HW+V
% U (відсутній) - вхід Y (позначення в розділі 4 xi*) -
вихід=поточний координаті ОБ
% X (Z) - внутрішній стан системи координата швидкість
% W (Lamnda Greek) - збурення в системі швидкості прискорення -
це ще 2 входи

```



```

% V (Delta Greek x) - шум вимірювань виходу тобто поточної
координати ОВ
Z0=[x0; 0];          % початковий стан системи координата швидкість
A=[0 1              % матриці системи в просторі стану
  0 0];
  B=[1 0            % 1 0 - задання швидкості ОВ
    0 0];          % 0 0
  G=[1 0           % G (P Greek) - вхідні збурення в системі - швидкість
    0 0];
C=[1 0];           % H - матриця вимірювань формує вихід з першого
елементу вектора стану тобто поточної координати
D=[0 0];           % вхід не впливає на вихід
H=[0 0];           % збурення на вихід не діють
[t1M,t1N] = size(t1);
U=zeros(t1M,2);    % вхід=0
W=zeros(t1M,2);    % Lamnda Greek=0 - збурення в системі
швидкість і прискорення
V=zeros(t1M,1);    % Delta Greek xi шум вимірювань координати
OV_CTime=ss(A,[B G],C,[D H]); % ОВ в просторі стану

% Моделювання неперервної системи 2-го порядку
if Type_V==1        % вхідна швидкість
  U(:,1)=v; U(:,2)=0; % швидкість та прискорення
end;
if Type_V==2        % вхідна швидкість
  U(:,1)=v; U(1:1000,1)=0; U(:,1)=lsim(tf([1],[Tu^2 2*Tu
1]),U(:,1),t1);
end;
if Type_V==3        % вхідна швидкість
  U(:,1)=v; U(1:500,1)=0; U(2000:2500,1)=0;
  for iii=1:500 U(iii,1)=v*iii/500; end;
  for iii=1:500 U(iii+2000,1)=v*(500-iii)/500; end;
  U(:,1)=lsim(tf([1],[Tu^2 2*Tu 1]),U(:,1),t1);
end;
for i=2:t1M-1
  U(i,2)=(U(i+1,1)-U(i-1,1))/2/dt1; % прискорення
end;
U(t1M,2)=U(t1M-1,2);
W(:,1)=randn(t1M,1).*(Sigma_Dist_v); % Lamnda Greek=0 - збурення
в системі швидкість
SysWx=tf([1],[TWx 1]);
Wx=lsim(SysWx,randn(t1M,1).*(Sigma_Dist_x),t1);
SysWv=tf([1],[TWv 1]); W(:,1)=lsim(SysWv,W(:,1),t1);
V=randn(t1M,1).*(SigmaNoise)+Wx; % шум вимірювань координати і
збурення координати
[Y,t2,X]=lsim(OV_CTime,[U W],t1,Z0); % моделювання - вихід
(координата) і вектор стану (координата і швидкість)
% figure; plot(t1,W(:,1),'g', t1,V,'m');
% title('Noise CTime System: W1=v g, V m');

```

```

% figure; plot(t2,Y,'k', t2,Y+V,'g', t2,U(:,1),'k',
t2,U(:,1)+W(:,1),'g'); % для рівномірного руху
% title('Output and State Vector CTime System: Y=x, k Y+V=x+v g,
U1=v k, U1+W1=v+w g');

% Математична модель для вимірювань за відеозображеннями
% дискретна система в просторі стану
%  $x[n+1] = Ax[n] + Bu[n] + Gw[n]$ 
%  $y[n] = Cx[n] + Du[n] + Hw[n] + v[n]$ 
% U (відсутній) - вхід=0 Y (позначення в розділі 4 xi*) -
вихід=поточній координаті OB
% X (Z) - внутрішній стан системи координати швидкість
прискорення
% W (Lamnda Greek) - збурення в системі координати швидкості
прискорення - це ще 3 входи додатково до U
% V (Delta Greek x) - шум вимірювань виходу тобто поточної
координати OB
% початковий стан системи координата швидкість прискорення
ZD0=zeros(1,FK_P);
for i=1:FK_P
    ZD0(i)=PARAM(i);
end;
% матриці системи в просторі стану
% A (F Greek) - неперервна модель рівноприскореного руху OB
A_=zeros(FK_P, FK_P);
for i=1:FK_P
    for j=i:FK_P
        A_(i,j)=dt1^(j-i)/factorial(j-i);
    end;
end;
BD=zeros(FK_P,1); % вхід відсутній
GD=A_; % G (P Greek) - передача збурень в системі
CD=zeros(1,FK_P); CD(1)=1; % H - матриця вимірювань формує
вихід з першого елемента вектора стану тобто поточної координати
DD=[0]; % вхід відсутній
HD=zeros(1,FK_P); % збурення на вихід не діють
[t1M,t1N] = size(t1);

% Дискретний фільтр Калмана для відеозображень
ZD=zeros(FK_P,t1M); GI=zeros(FK_P,t1M); P=zeros(FK_P,FK_P,t1M);
I3_3=zeros(FK_P,FK_P);
for i=1:FK_P
    I3_3(i,i)=1;
end;
ZD(:,1)=ZD0;
QN=zeros(FK_P,FK_P);
if FK_P>=1 QN(1,1)=Sigma_Dist_x^2; end;
if FK_P>=2 QN(2,2)=Sigma_Dist_v^2; end;
if FK_P>=3 QN(3,3)=Sigma_Dist_a^2; end;
P=zeros(FK_P,FK_P);
for i=1:FK_P

```

```

        for j=1:FK_P
            P(i,j)=ZD(i)*ZD(j);
        end;
    end;
    if FK_P>=1 P(1,1)=P(1,1)+Sigma_Dist_x0^2; end;
    if FK_P>=2 P(2,2)=P(2,2)+Sigma_Dist_v0^2; end;
    if FK_P>=3 P(3,3)=P(3,3)+Sigma_Dist_a0^2; end;
    for i=2:t1M
        Pil=A_*P(:, :, (i-1))*A_'+GD*QN*GD';
        GI(:,i)=Pil*CD'*(CD*Pil*CD'+SigmaNoise^2)^(-1);
        ZD(:,i)=A_*ZD(:,i-1)+GI(:,i)*(Y(i)+V(i)-CD*(A_*ZD(:,i-1)));
        P(:, :, i)=Pil*(I3_3-GI(:,i)*CD);
    end;

% експоненційне згладжування
ZDEKS=zeros(3,t1M);
for i=EKS_P+1:t1M
    ZDEKS(1,i)=(1-ksi)*(Y(i)+V(i));% згладжування по координаті
%ZDEKS(2,i)=(1-ksi)*ZDDIFRV(2,i,jjj);%згладжування по швидкості
%ZDEKS(3,i)=(1-ksi)*ZDDIFRV(3,i,jjj);%згладжування по
прискоренню

    for j=1:EKS_P
        ZDEKS(:,i)=ZDEKS(:,i)-ksi.*(-1).^j.*EKS_P./j.*ZDEKS(:,i-
j);
    end;
end;

% Оцінка похибок
DZDFull=zeros(3,t1M,size(Noc,1));
DZDEKS=zeros(3,t1M);

for j=1:size(Noc,1)
    DZDFull(1,:,j)=ZD(1,:)-Y'; DZDFull(2,:,j)=ZD(2,:)-U(:,1)';
    DZDFull(3,:,j)=ZD(3,:)-U(:,2)';
end;
DZDEKS(1,:)=ZDEKS(1,:)-Y'; DZDEKS(2,:)=ZDEKS(2,:)-U(:,1)';
DZDEKS(3,:)=ZDEKS(3,:)-U(:,2)';

if Type_V==1
    DZD=zeros(3,FKMean+1,size(Noc,1));
    for j=1:size(Noc,1)
        DZD(:, :, j)=DZDFull(:,Noc(j):Noc(j)+FKMean,j);
    end;
else
    DZD=zeros(3,size(U,1)-510,size(Noc,1));
    for j=1:size(Noc,1)
        DZD(:, :, j)=DZDFull(:,Noc(j):size(U,1)-510+Noc(j)-1,j);
    end;
    DZD=DZDFull(:,510:size(U,1)-510,:);
end;
end;

```

```

DZDEKS=DZDEKS(:,510:size(U,1)-510);
DZD_Max1=zeros(size(Noc,1),1); DZD_Max2=zeros(size(Noc,1),1);
DZD_Max3=zeros(size(Noc,1),1);
DZD_Std1=zeros(size(Noc,1),1); DZD_Std2=zeros(size(Noc,1),1);
DZD_Std3=zeros(size(Noc,1),1);
for j=1:size(Noc,1)
    DZD_Max1(j)=max(abs(DZD(1, :, j)));
    DZD_Max2(j)=max(abs(DZD(2, :, j)));
    DZD_Max3(j)=max(abs(DZD(3, :, j)));
    DZD_Std1(j)=std(DZD(1, :, j)); DZD_Std2(j)=std(DZD(2, :, j));
    DZD_Std3(j)=std(DZD(3, :, j));
end;

DZDEKS_Max1=zeros(size(Noc,1),1);
DZDEKS_Max2=zeros(size(Noc,1),1);
DZDEKS_Max3=zeros(size(Noc,1),1);
DZDEKS_Std1=zeros(size(Noc,1),1);
DZDEKS_Std2=zeros(size(Noc,1),1);
DZDEKS_Std3=zeros(size(Noc,1),1);
DZDEKS_Max1(:)=max(DZDEKS(1, :));
DZDEKS_Max2(:)=max(DZDEKS(2, :));
DZDEKS_Max3(:)=max(DZDEKS(3, :));
DZDEKS_Std1(:)=std(DZDEKS(1, :));
DZDEKS_Std2(:)=std(DZDEKS(2, :));
DZDEKS_Std3(:)=std(DZDEKS(3, :));

% Теоретичний розрахунок похибок
% Фільтр Калмана
DZD_StdTx=zeros(size(Noc,1),1); DZD_StdTv=zeros(size(Noc,1),1);
DZD_StdTx(:)=sqrt(SigmaNoise*dt1*sqrt(2*Sigma_Dist_a*SigmaNoise+
Sigma_Dist_v^2));
DZD_StdTv(:)=sqrt(Sigma_Dist_a*dt1*sqrt(2*Sigma_Dist_a*SigmaNoise
e+Sigma_Dist_v^2));
%DZD_StdTa(:)=0;
figure; % графіки параметрів руху для Noc=jjj
plot(t1,Y,'b',t1,Y+V,'g',t1,ZD(1,:), 'r',t1,ZDEKS(1,:), 'c');
xlabel('Час, с'); ylabel('Координата, мм');
grid on;

% графіки похибок параметрів руху
figure;
plot(Noc*dt1,DZDEKS_Std1,'m-',Noc*dt1,DZD_Std1,'b-
.', 'LineWidth',3);
xlabel('Кількість відліків координати'); ylabel('СКЗ похибки,
мм');
grid on;

```

СПИСОК ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. Тищенко Н. М. Введение в проектирование систем управления / Н. М. Тищенко. – 2-е изд., перераб. и доп. – М.: Энергоатомиздат, 1986. – 248 с.
2. Автоматизированное проектирование систем управления / Под ред. М. Джамшиди и др. ; пер. с англ. В. Г. Дунаева и А. Н. Косилова. – М. : Машиностроение, 1989. – 344 с.
3. Хетагуров Я. А. Проектирование автоматизированных систем обработки информации и управления (АСОУИ) / Я. А. Хетагуров. – М. : Высшая школа, 2006. – 223 с.
4. Автоматизированное проектирование систем автоматического управления / Я. Я. Алексакин, А. Е. Бржозовский, В. А. Жданов и др. : под ред. В. В. Солодовникова – М. : Машиностроение, 1990. – 332 с.
5. Таланчук П. М. Основы теории и проектирования измерительных приборов : учебное пособие / П. М. Таланчук, В. Т. Рущенко. – К. : Вища школа, 1989. – 454 с.
6. Солодовников В. В. Основы теории и элементы систем автоматического регулирования : учебное пособие для вузов / В. В. Солодовников, В. Н. Плотников, А. В. Яковлев. – М. : Машиностроение, 1985. – 536 с.
7. Глушков В. М. Синтез цифровых автоматов / В. М. Глушков. – М. : Физматгиз, 1962. – 476 с.
8. Блейкли Т. Р. Проектирование цифровых устройств с малыми и большими интегральными схемами / Т. Р. Блейкли : пер. с англ. ; под ред. А. И. Петренко. – К. : Вища школа. – 1981. – 336 с.
9. Корнійчук А. І. Методика складання рівнянь управління логічних об'єктів / А. І. Корнійчук. – Житомир : ЖІТІ, 1996. – 196 с.
10. Лазарев В. Г. Синтез управляющих автоматов / В. Г. Лазарев, Е. И. Пийль. – 2-е изд., перераб. и доп. – М. : Энергия, 1978. – 408 с.
11. Кузнецов О. Дискретная математика для инженера : учебное пособие / О. Кузнецов. – 6-е изд., стереотипное. – СПб. : Издательство «Лань», 2014. – 400 с.
12. Горбатов В. А. Семантическая теория проектирования автоматов / В. А. Горбатов. – М. : Энергия, 1979. – 264 с.
13. Фридман А. Теория и проектирование переключательных схем / А. Фридман, П. Менон. – М. : Мир, 1978. – 580 с.
14. Кравцов А. Ф. Расчет автоматических систем контроля и регулирования металлургических процессов / А. Ф. Кравцов и др. – К. : Высшая школа, 1981. – 320 с.
15. Захаров Н. Г. Синтез цифровых автоматов : учебное пособие / Н. Г. Захаров, В. Н. Рогов. – Ульяновск : УлГТУ, 2003. – 250 с.
16. Каган Б. М. Электронные вычислительные машины и системы / Б. М. Каган. – М. : Энергоатомиздат, 1991. – 592 с.

17. Гилмор Ч. Введение в микропроцессорную технику / Ч. Гилмор. – М. : Мир, 1984. – 334 с.
18. Баранов С. И. Синтез микропрограммных автоматов (граф-схемы и автоматы). – 2-е изд., перераб. и доп. / С. И. Баранов. – Л. : Энергия, 1979. – 232 с.
19. Баранов С. И. Микропрограммирование: принципы, методы, применения / С. И. Баранов, А. А. Баркалов // Зарубежная радиоэлектроника. – 1984. – № 5. – С. 3–29.
20. Колосов В. Г. Проектирование узлов и систем автоматики / В. Г. Колосов, В. Ф. Мелехин. – Л. : Энергоатомиздат, 1983. – 256 с.
21. Сарвин А. А. Диагностика и надежность автоматизированных систем / А. А. Сарвин, Л. И. Абакулина, О. А. Готшалк. – СПб. : СЗТУ, 2003. – 69 с.
22. Гуляев В. А. Диагностика вычислительных машин / В. А. Гуляев и др. – К. : Техніка, 1981. – 167 с.
23. Иыуду К. А. Надежность, контроль и диагностика вычислительных машин и систем / К. А. Иыуду. – М.: Высшая школа, 1989. – 216 с.
24. Гостев В. І. Оптималне керування складними об'єктами / В. І. Гостев, О. А. Баранов, І. П. Лісовий, С. О. Шматок. – К. : Радіоаматор, 2000. – 208 с.
25. Ерофеев А. А. Теория автоматического управления : учебник для вузов / А. А. Ерофеев. – СПб. : Политехника, 2003. – 302 с.
26. Гостев В. Проектирование нечетких регуляторов для систем автоматического управления / В. Гостев. – СПб. : ВHV-СПб, 2011. – 416 с.
27. Кузьмин С. З. Основы проектирования систем цифровой обработки радиолокационной информации / С. З. Кузьмин. – М. : Радио и связь, 1986.-352с.
28. Фалькович С. Е. Статистическая теория измерительных радиосистем / С. Е. Фалькович, Э. Н. Хомяков. – М. : Радио и связь, 1981. – 288 с.
29. Справочник по теории автоматического управления / под ред. А. А. Красовского. – М. : Наука, 1987. – 712 с.
30. Никульчев Е. В. Практикум по теории управления в среде MATLAB : учебное пособие / Е. В. Никульчев. – М. : МГАПИ, 2002. – 88 с.
31. Пат. 103422 С2 Україна, МПК (2013.01)G01В 7/00, G01Р 3/36 (2006.01). Пристрій для вимірювання параметрів руху об'єктів / Подчашинський Ю. О., Шаповалова (Лугових) О. О. ; заявник і власник патенту ЖДТУ. – № а201208343 ; заявл. 07.07.2012 ; опубл. 10.10.2013, Бюл. № 9.
32. Форсайт Д. Компьютерное зрение. Современный подход / Д. Форсайт, Ж. Понс. – М. : Вильямс, 2004. – 928 с.
33. Кузовков Н. Т. Непрерывные и дискретные системы управления и методы идентификации / Н. Т. Кузовков, С. В. Карабанов, О. С. Салычев. – М. : Машиностроение, 1978. – 222 с.
34. Кузовков Н. Т. Инерциальная навигация и оптимальная фильтрация / Н. Т. Кузовков, О. С. Салычев. – М. : Машиностроение, 1982. – 216 с.

35. Цыпкин Я. З. Информационная теория идентификации / Я. З. Цыпкин. – М. : Наука, 1995. – 336 с.
36. Балакришнан А. Теория фильтрации Калмана : пер. с англ. / А. Балакришнан. – М. : Мир, 1988. – 168 с.
37. Браммер К. Фильтр Калмана-Бьюси / К. Браммер, Г. Зиффилинг. – М. : Наука, 1982. – 189 с.
38. Сеницын И. Н. Фильтры Калмана и Пугачева : учебное пособие / И. Н. Сеницын. – М. : Университетская книга ; Логос, 2006. – 640 с.
39. Автоматизированное проектирование систем автоматического управления / Я. Я. Алексанкин, А. Э. Бржозовский, В. А. Жданов и др. ; под ред. В. В. Солодовникова. – М. : Машиностроение, 1990. – 332 с.
40. Сизиков В. С. Устойчивые методы обработки результатов измерений : учебное пособие / В. С. Сизиков. – СПб. : СпецЛит, 1999. – 240 с.
41. Васильев К. К. Методы обработки сигналов : учебное пособие / К. К. Васильев. – Ульяновск : УлГТУ, 2001. – 80 с.
42. Кузовков Н. Т. Модальное управление и наблюдающие устройства / Н. Т. Кузовков. – М. : Машиностроение, 1976. – 184 с.
43. Арсеньев Г. Н. Радиоавтоматика. Ч. 1. Теория линейных непрерывных систем автоматического управления РЭС : учебник для вузов / Г. Н. Арсеньев, Г. Ф. Зайцев. – М. : Сайнс-Пресс, 2008. – 408 с.
44. Радиоавтоматика : учебное пособие / В. А. Бесекерский, А. А. Елисеев, А. В. Небылов и др. ; под ред. В. А. Бесекерского. – М. : Высшая школа, 1985. – 271 с.
45. Болнокин В. Е. Анализ и синтез систем автоматического управления на ЭВМ. Алгоритмы и программы / В. Е. Болнокин, П. И. Чинаев. – М. : Радио и связь, 1986. – 248 с.
46. Саматокін Б.Б. Лекції з теорії автоматичного керування : навчальний посібник для студентів вищих навчальних закладів. – Житомир : ЖІТІ, 2001. – 508 с.
47. Тютюник А.Г. Оптимальні і адаптивні системи автоматичного керування. Практикум : навчальний посібник. – Житомир : ЖІТІ, 2002. – 424 с.
48. Шматок С.О., Подчашинський Ю.О. Автоматизоване проектування систем керування на основі MATLAB. Навчальний посібник. – Житомир : ЖДТУ, 2005. – 172 с.
49. Шматок С. О. Математичні та програмні засоби моделювання пристроїв і систем управління. Використання нечітких множин та нейронних мереж : навчальний посібник / С. О. Шматок, Ю. О. Подчашинський, О. С. Шматок. – Житомир : ЖДТУ, 2007. – 280 с.
50. Безвесільна О. М. Методи оптимізації цільової функції та ідентифікації характеристик прецизійних навігаційних систем : монографія / О. М. Безвесільна, Ю. В. Киричук, Ю. О. Подчашинський. – Житомир : ЖДТУ, 2010. – 201 с.

Навчальне видання

ПОДЧАШИНСЬКИЙ Юрій Олександрович
ШАВУРСЬКИЙ Юрій Олександрович
ЛУГОВИХ Оксана Олександрівна

**ПРОЕКТУВАННЯ ТА КОНСТРУЮВАННЯ ПРИСТРОЇВ
ТА СИСТЕМ УПРАВЛІННЯ**

Навчальний посібник

Підписано до друку 02.10.18. Формат 60x84¹/₁₆. Папір офсетний.
Гарнітура Times New Roman. Умов. друк. аркушів 16,28.
Наклад 300 прим. Зам. № 348.

Редакційно-видавничий сектор бібліотеки
Житомирського державного технологічного університету
вул. Чуднівська, 103, м. Житомир, 10005

Свідоцтво про внесення до Державного реєстру суб'єкта видавничої справи
Серія ЖТ № 08 від 26.03.2004