

Ловейкін В.С., Ромасевич Ю.О., Човнюк Ю.В.

МЕХАТРОНІКА

НАВЧАЛЬНИЙ ПОСІБНИК

Київ – 2012

УДК 621.396.6

Публікується за рішенням вченої ради технічного навчально-наукового інституту Національного університету біоресурсів і природокористування України (протокол №2 від 22 жовтня 2012 року).

Рецензенти:

Сукач Михайло Кузьмич, доктор технічних наук, професор, професор кафедри будівельних машин Київського національного університету будівництва і архітектури.

Мироненко Валентин Григорович, доктор технічних наук, професор, Національний науковий центр „Інститут механізації та електрифікації сільського господарства” Національної академії аграрних наук України

Ловейкін В.С., Ромасевич Ю.О., Човнюк Ю.В.
Мехатроніка. Навчальний посібник. – К., 2012. - 357 с.

У навчальному посібнику викладено основні положення проектування та експлуатації мехатронних систем. Значна увага приділена розробці апаратної частини мехатронних об'єктів, до якої належать датчики, приводи, інтерфейсні та обчислювальні пристрої. Описані основні положення синтезу режимів руху механізмів мехатронних систем та організації взаємодії між їх елементами.

Навчальний посібник призначений для студентів вищих навчальних закладів, які навчаються за спеціальностями 8.090219 „Обладнання лісового комплексу”, 8.090215 „Машини і обладнання сільськогосподарського виробництва”, 8.0919.01 „Механізація сільського господарства” а також може бути корисним для аспірантів і слухачів факультетів підвищення кваліфікації.

ВСТУП

Сучасний термін „Мехатроніка” („Mechatronics”), згідно із японським джерелом [1], був уведений фірмою Yaskawa Electric в 1969 році й зареєстрований як торговельна марка в 1972 році. Ця назва отримана комбінацією слів „МЕХАніка” й „ЕлекТРОНІКА”. Об'єднання цих понять у єдиному словосполученні означає інтеграцію знань в відповідних галузях науки й техніки, що дозволила зробити якісний стрибок у створенні техніки нових поколінь і виробництві новітніх видів систем й обладнання.

Аналогічним чином йшов розвиток електромеханіки як науки, що використовує досягнення електротехніки й механіки при створенні приводних виконавчих систем широкого призначення. Інтеграція електромеханіки й мікроелектроніки призвела до появи комплектних інтегрованих мехатронних модулів руху робочих органів та вузлів машин, а також створюваного на їхній основі обладнання [2].

Однак до початку 80-х років термін „Мехатроніка” затверджується у світовій технічній літературі як назва цілого класу машин з комп'ютерним керуванням рухом.

Тут інтегруються досягнення: електромеханіки + електроніки + систем комп'ютерного керування рухами машин і складних просторових механізмів.

В „Oxford Illustrated Encyclopedia” [3] можна прочитати: *мехатроніка* - японський термін для опису технологій, що виникли на стику електротехніки, машинобудування й програмного забезпечення. Включає проектування, виробництво й вивчає функціонування машин з „розумною” поведінкою, тобто діючих по заданій програмі, їхнього зв'язку з іншими системами (штучний інтелект, вимірювальне обладнання, систем керування).

Мехатроніка – це нова область науки і техніки, присвячена створенню й експлуатації машин та систем з комп'ютерним керуванням рухом, котра базується на знаннях у області механіки, електроніки та мікропроцесорної техніки, інформатики й комп'ютерного керування рухом машин та агрегатів [4].

У цьому визначенні підкреслена триєдина сутність мехатронних систем, у основу побудови котрих закладена ідея глибокої взаємної залежності механічних+електронних+комп'ютерних елементів [5, 6].

(перетин площин)

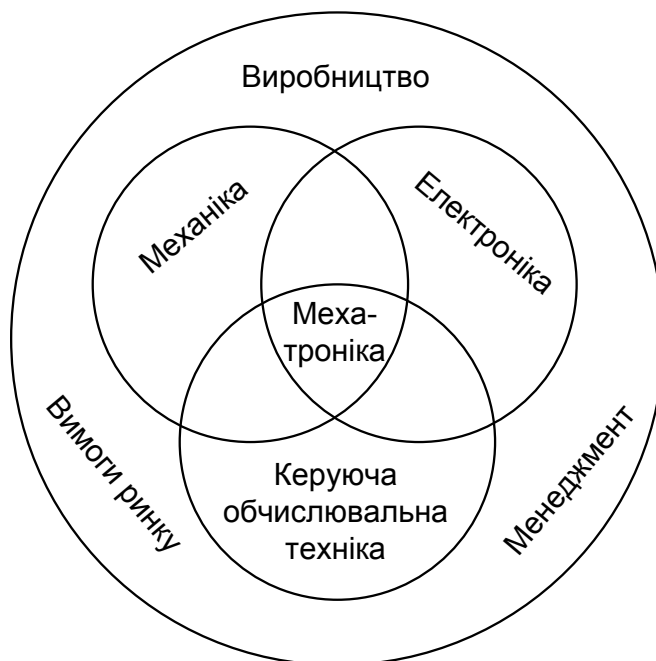


Рис. 1. Визначення мехатронних систем

Отже системна інтеграція трьох вказаних видів елементів є необхідною умовою побудови мехатронних систем [7-9].

Мехатронні системи – це технічні системи, що забезпечують виробничі процеси, засновані на комплексному використанні електричних, механічних та інформаційних технологій (рис. 2).

Ґрунтуючись на викладеному вище, визначимо теоретичну основу мехатронних систем як сукупність методів формалізації і математичного опису, прийнятих у механіці, електротехніці й інформатиці [10]. Основним засобом опису таких систем є використання диференціального й інтегрального числення, цифрових методів. У мехатроніці математичний апарат від прикладної механіки до цифрових методів становить об'єднання різних математичних методів і засобів. Аналіз мехатронних систем припускає послідовність їх узагальненого, графічного, аналітичного й

алгоритмічного опису. Аналіз і синтез мехатронних систем пов'язаний з узагальненням і розробкою єдиного математичного апарату алгоритмізації процесів взаємодії їхніх елементів.



Рис.2. Зв'язок мехатроніки з іншими предметами

Розглянемо узагальнену структуру машин з комп'ютерним керуванням, орієнтованих на завдання автоматизованого машинобудування, що представлена на рис. 3. Ця структура автоматичних роботів, уведена академіком Е.П. Поповим, лежить у основі побудови даної схеми [11].

Зовнішнім середовищем для машин розглядуваного класу є технологічне середовище, яке складається з різноманітного основного та допоміжного обладнання, технологічного оснащення й об'єктів робіт.

При виконанні мехатронною системою заданого функціонального руху об'єкти робіт справляють збурюючий вплив на робочий орган. Приклади таких впливів: а) сили різання для операцій механообробки; б) контактні

сили й моменти сил при складанні; в) сила реакції струменя рідини при операції гідравлічного різання.

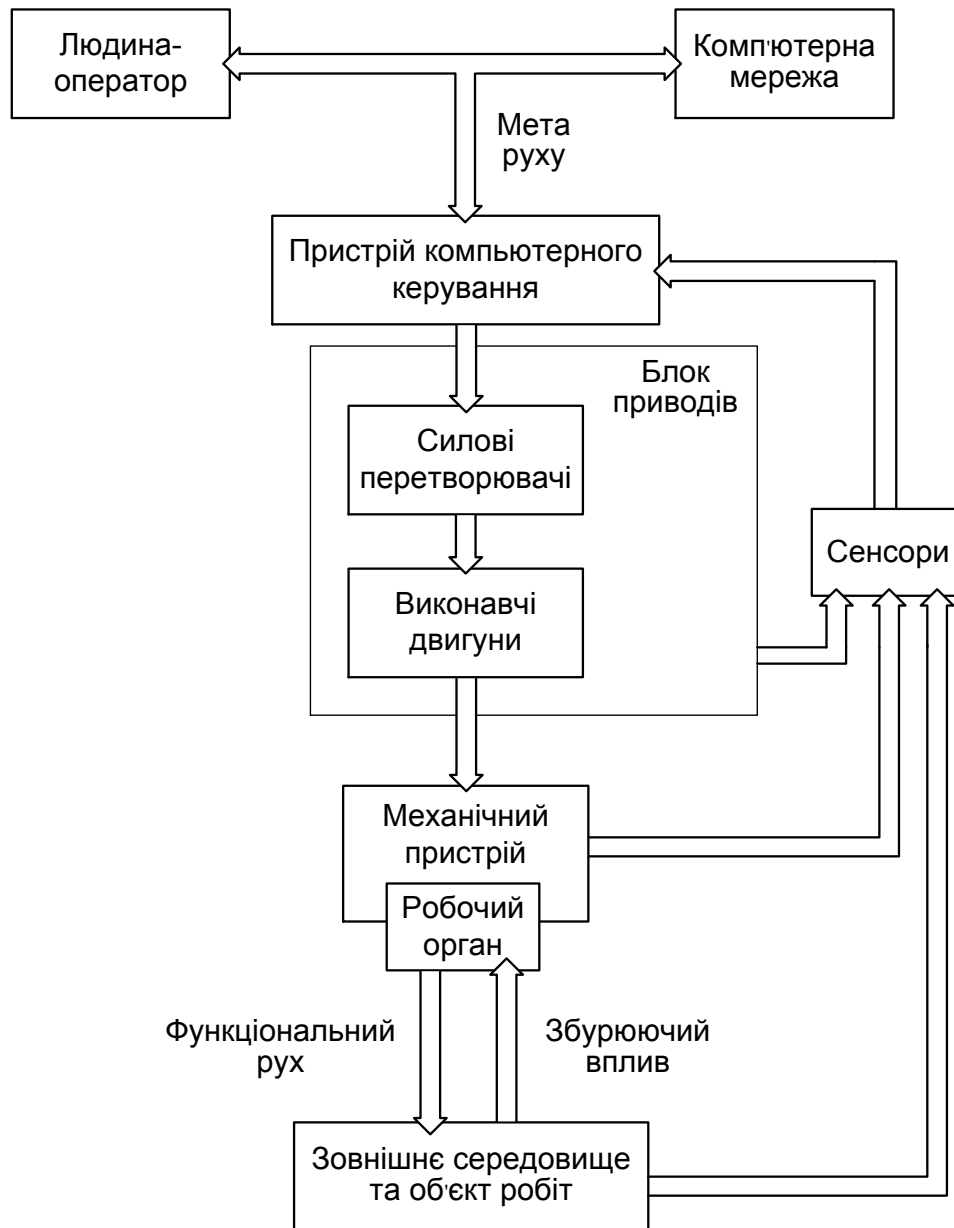


Рис. 3. Узагальнена схема машин з комп'ютерним керуванням руху

Зовнішні середовища поділяються на два основних класи: 1) детерміновані; 2) недетерміновані.

До *детермінованих* відносяться середовища, для яких параметри збурюючих впливів і характеристики об'єктів робіт можуть бути заздалегідь визначені з необхідним для проектування мехатронної системи величиною/ступенем точності.

Деякі середовища є *недермінованими* за своєю природою (наприклад, екстремальні середовища: підводні, підземні тощо).

Характеристики технологічних середовищ, як правило, можуть бути визначені за допомогою аналітико-експериментальних досліджень і методів комп'ютерного моделювання. Наприклад, для оцінки сил різання при механообробці проводять серії експериментів на спеціальних дослідних установках, параметри вібраційних впливів вимірюють на вібростендах з наступним формуванням математичних і комп'ютерних моделей збурюючих впливів на основі експериментальних даних. Однак для організації й проведення подібних досліджень зазвичай потрібні занадто складні й великої вартості апаратура та вимірювальні технології. Тому у таких випадках доцільно застосовувати *методи адаптивного керування*, які дозволяють автоматично корегувати закон руху мехатронної системи безпосередньо у ході виконання операції [12].

До складу машини з комп'ютерним керуванням рухом входять наступні основні компоненти: 1) механічний пристрій, кінцевим ланцюгом якого є робочий орган; 2) блок приводів, що включає силові перетворювачі й виконавчі двигуни; 3) пристрій комп'ютерного керування, верхнім рівнем для котрого є людина-оператор, або інша ЕОМ, що входить в комп'ютерну мережу; 4) сенсори, призначені для передачі в пристрій керування інформації про фактичний стан блоків машини й рух мехатронної системи.

Таким чином, наявність трьох обов'язкових частин – 1) механічної (точніше електромеханічної); 2) електронної 3) комп'ютерної, пов'язаних енергетичними й інформаційними потоками, є первинною ознакою, яка відрізняє мехатронні системи.

Електромеханічна частина включає механічні ланцюги та передачі, робочий орган, електродвигуни, сенсори й додаткові електротехнічні елементи (гальма, муфти).

Механічний пристрій призначений для перетворення рухів ланцюгів у необхідний рух робочого органа.

Електронна частина складається з мікроелектронних пристроїв, силових перетворювачів й електроніки вимірювальних ланцюгів.

Сенсори призначені для збору даних про фактичний стан зовнішнього середовища та об'єктів робіт, механічного пристрою й блоку приводів з наступною первинною обробкою й передачею цієї інформації в *пристрій комп'ютерного керування*. До складу пристрою комп'ютерного керування мехатронної системи зазвичай входять: а) комп'ютер верхнього рівня; б) контролери керування рухом.

Функції пристрою комп'ютерного керування (основні) [4]:

1. Керування процесом механічного руху мехатронного модуля або багатовимірної системи в реальному часі з обробкою сенсорної інформації.
2. Організація керування функціональними рухами мехатронної системи, котра передбачає координацію керування механічним рухом мехатронної системи й супутніми зовнішніми процесами. Як правило, для реалізації функції керування зовнішніми процесами використовуються дискретні входи/виходи пристрою.
3. Взаємодія з людиною-оператором через людино-машинний інтерфейс у режимах автономного програмування (режим *off-line*) і безпосередньо в процесі руху мехатронної системи (режим *on-line*).
4. Організація обміну даними з периферійними пристроями, сенсорами й іншими пристроями системи.

Задачою мехатронної системи є перетворення вхідної інформації, що надходить із верхнього рівня керування, у цілеспрямований механічний рух з керуванням на основі принципу зворотного зв'язку. Характерно, що електрична енергія (рідше гідравлічна або пневматична) використовується в сучасних системах як проміжна енергетична форма.

Прикладом мехатронної системи є система керування приводом маніпулятора промислового робота, структурна схема якої показана на рис. 4. Основними елементами системи є підсилювач постійного струму,

серводвигун постійного струму, а також потенціометр, який використовується для формування сигналу зворотного зв'язку. Із рис. 4 видно, що сигнал величини кута α_1 повороту маніпулятора задає мікро-ЕОМ. Сигнал величини α_1 подається на перший вхід підсилювача постійного струму, а на другий вхід – сигнал зворотного зв'язку, сформований потенціометром за величиною реального кута повороту маніпулятора α_2 . Різниця сигналів підсилюється та надходить в обмотку серводвигуна постійного струму, змушуючи його обертатися. Обертання серводвигуна викликає відповідне обертання маніпулятора, причому швидкість руху маніпулятора зменшується за рахунок установки редуктора між серводвигуном і поворотним механізмом маніпулятора. Поточне значення кута повороту маніпулятора перетворюється потенціометром у відповідний сигнал зворотного зв'язку, віднімається від сигналу кута задання та різниця знову надходить у підсилювач постійного струму. Таким чином, поточне значення кута α_2 поступово встановлюється на рівні кута, який задається мікро-ЕОМ. Загалом система має елементи механіки, електроніки та інформатики і тому вона є мехатронною.

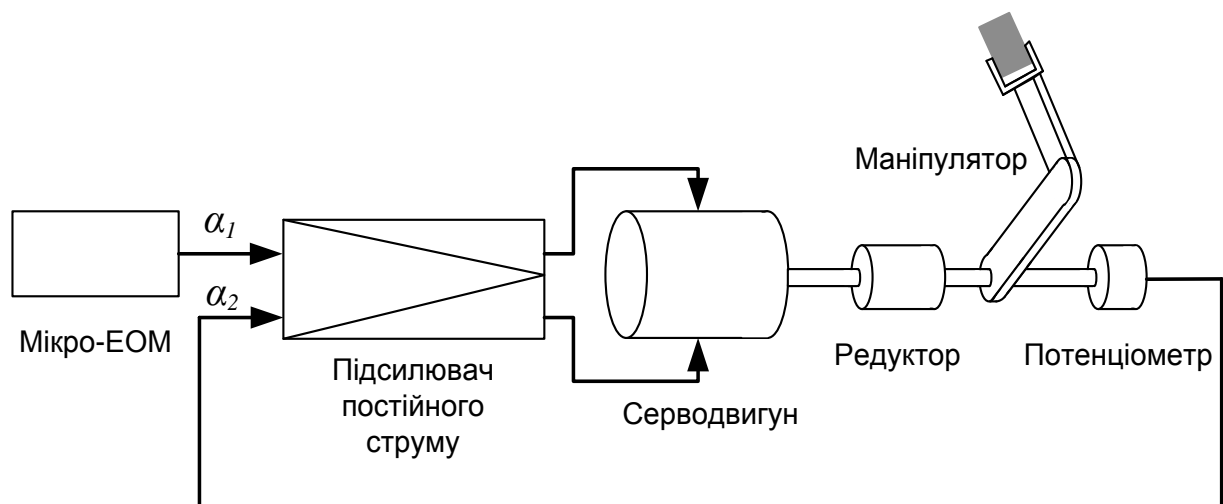


Рис. 4. Структурна схема системи керування маніпулятором

Сутність мехатронного підходу до проектування полягає у інтеграції в єдиний функціональний модуль двох або більше елементів можливо навіть різної фізичної природи. Іншими словами, на стадії проектування з

традиційної структури машини виключають як сепаратний (окремий) пристрій принаймні один інтерфейс при збереженні фізичної сутності перетворення, яке виконує даний модуль.

В ідеальному варіанті мехатронний модуль, одержавши на вхід інформацію про мету керування, буде виконувати з бажаними показниками якості заданий функціональний рух. Апаратне об'єднання елементів у єдині конструктивні модулі повинно обов'язково супроводжуватися розробкою інтегрованого програмного забезпечення.

Програмні засоби мехатронних систем повинні забезпечувати безпосередній перехід від задумки системи через її математичне моделювання до керування функціональним рухом у реальному часі.

При створенні машин з комп'ютерним керуванням мехатронний підхід дає наступні переваги у порівнянні з традиційними засобами автоматизації:

- відносно низьку вартість завдяки високому ступеню інтеграції, уніфікації й стандартизації всіх елементів й інтерфейсів;
- високу якість реалізації складних і точних рухів внаслідок застосування методів інтелектуального керування;
- високу надійність, довговічність та захищеність від похибок / помилок ззовні;
- конструктивну компактність модулів (до мініатюризації в мікромашинах);
- поліпшені масогабаритні й динамічні характеристики машин внаслідок спрощення кінематичних ланцюгів;
- можливість комплектування функціональних модулів у складні системи й комплекси під конкретні завдання замовника.

Сьогодні мехатронні модулі й системи знаходять широке застосування в наступних областях:

- ▶ верстатобудування й устаткування для автоматизації технологічних процесів;
- ▶ робототехніка (промислова та спеціальна);

- ▶ авіаційна, космічна та військова техніка;
- ▶ автомобілебудування (наприклад, антиблокувальні системи гальм, системи стабілізації руху автомобіля й автоматичного паркування);
- ▶ нетрадиційні транспортні засоби (електровелосипеди, вантажні візки, електроролери, інвалідні візки);
- ▶ офісна техніка (наприклад, копіювальні й факсимільні апарати);
- ▶ елементи обчислювальної техніки (наприклад, принтери, плотери, дисководи);
- ▶ медичне обладнання (реабілітаційне, клінічне, сервісне);
- ▶ побутова техніка (пральні, швейні, посудомийні та інші машини);
- ▶ мікромашини (для медицини, біотехнології, засобів телекомунікації);
- ▶ контрольні-вимірювальні пристрої і машини;
- ▶ фото- і відеотехніка;
- ▶ тренажери для підготовки пілотів й операторів ;
- ▶ шоу-індустрія (системи звукового й світлового оформлення).

Можна виділити наступні тенденції зміни й ключові вимоги світового ринку в розглянутій області:

- 1) необхідність випуску й сервісу обладнання у відповідності з міжнародною системою стандартів якості, сформульованих у стандарті ISO 9000;
- 2) інтернаціоналізація ринку науково-технічної продукції й, як наслідок, необхідність активного впровадження в практику форм і методів міжнародного інжинірингу й трансферу технологій;
- 3) підвищення ролі малих і середніх виробничих підприємств в економіці завдяки їхній здатності до швидкого й гнучкого реагування на вимоги, що змінюються;
- 4) бурхливий розвиток комп'ютерних систем і технологій, засобів телекомунікації; прямим наслідком цієї загальної тенденції є інтелектуалізація систем керування механічним рухом і технологічними функціями сучасних машин.

1. ПРИСТРОЇ МЕХАТРОННИХ СИСТЕМ

1.1. Датчики

1.1.1. Класифікація датчиків

Датчик (сенсор) – пристрій, який призначений для перетворення інформації, що поступає на його вхід у вигляді деякої фізичної величини, в іншу функціональну величину, для подальшого використання в елементах мехатронних систем.

За точністю датчики повинні відповідати класам: 0,25; 0,4; 0,6; 1,0; 1,5; 2,5; 4,0. Датчики класифікуються за різними ознаками, але перед усім за видом величини, що вимірюється, та принципом дії (табл. 1.1) [13].

Таблиця 1.1. Типи датчиків та величини, які вони можуть вимірювати

Величина, що вимірюється	Тип датчика											
	потенціометр.	тензометричн.	індуктивні	терморезисторні	ємкісні	фоторезисторні	електронні	індукційні	п'єзоелектричні	термоелектричні	датчики Холла	фотоелектричні
1	2	3	4	5	6	7	8	9	10	11	12	13
Переміщення	+	+	+	-	+	-	+	-	-	-	+	+
Рівень	+	-	+	-	-	-	-	-	-	-	-	-
Швидкість	-	-	-	+	+	-	-	+	-	-	-	+
Прискорення	-	+	-	-	-	-	-	+	+	-	-	-
Сила	+	+	+	-	+	+	+	+	+	-	+	-
Тиск	-	+	+	-	+	-	+	-	+	-	+	-
Момент	+	+	+	-	+	+	-	-	-	-	-	+
Вологість	-	-	-	+	+	-	-	-	-	+	-	-
Температура	-	-	+	+	-	+	-	-	-	+	-	+

1	2	3	4	5	6	7	8	9	10	11	12	13
Витрата речовини	+	+	+	-	-	+	-	+	-	-	-	-
Вібрація	+	+	+	-	+	-	-	+	+	-	+	-

+ застосовується; - не застосовується.

Датчики перетворюють неелектричну величину в електричну або електричну в електричну, або неелектричну в неелектричну. Датчики за принципом дії поділяються на:

- *параметричні* (реагують на фізичний параметр) – до них відносять: контактні, реостатні, потенціометричні, тензодатчики, терморезистори, ємкісні, індуктивні, електронні, фоторезисторні та ін. Вимагають застосування додаткового джерела живлення;
- *генераторні* (виробляють фізичний сигнал) - до них відносять: термоелектричні (термопари), індукційні, п'єзоелектричні, вентильні фотоелементи та ін. Їм не потрібне додаткове джерело живлення.

1.1.2. Основні характеристики датчиків

До основних характеристик датчиків належать перераховані нижче.

1. Статична характеристика – залежність вихідної величини від вхідної $y=f(x)$.
2. Чутливість (коефіцієнт перетворення) – відношення вихідної величини до вхідної величини $k = y/x$.
3. Межа чутливості – мінімальна величина на вході датчика, що викликає зміну його вихідної величини (залежить від внутрішніх та зовнішніх факторів – тертя, люфту, гістерезисну, перешкод).
4. Похибка: абсолютна – різниця між дійсним значенням вихідної величини та розрахунковим значенням ($\Delta y = y' - y$); відносна - $\delta = \Delta y \cdot 100\%/y$.
5. Динамічна характеристика це залежність вихідної величини від вхідної в перехідний період.

6. Вихідна потужність.
7. Вихідний опір.

1.1.3. Види датчиків та принципи їх роботи

1.1.3.1. Електроконтактні датчики

Вони будуються на основі перетворювачів, які перетворюють механічне переміщення в замкнутий або розімкнутий стан контактів, які керують електричним колом (рис. 1.1). Існує велика кількість конструкцій цих датчиків різного призначення. Досить широке застосування знайшов двоконтактний датчик [14].

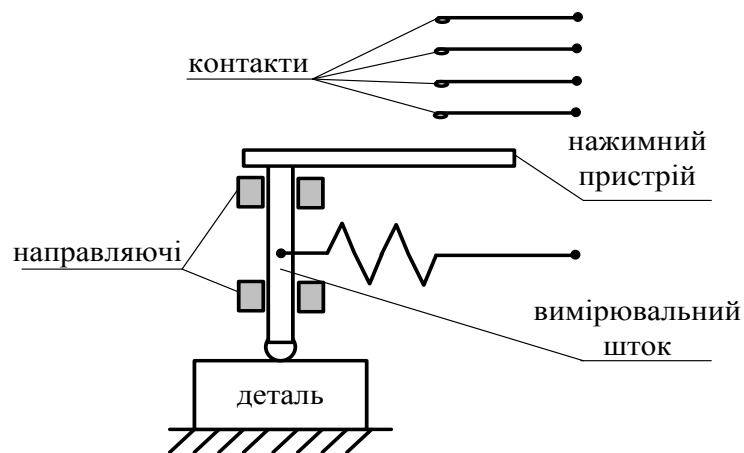


Рис. 1.1. Принцип дії та схема вмикання електроконтактного датчика

1.1.3.2. Потенціометричні датчики

Використовуються для перетворення лінійного (x) або кутового (φ) переміщення (вхідна величина) в електричний сигнал постійного або змінного струму (вихідна величина) (рис. 3.2). Конструктивно складаються з каркасу, обмотки та движка.

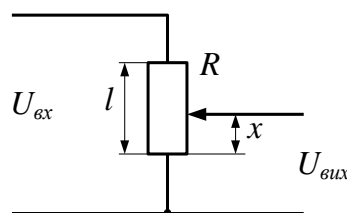


Рис. 1.2. Принцип дії та схема вмикання потенціометричного датчика

Вихідний сигнал датчика визначається такою формулою:

$$U_{\text{в\ddot{e}д}} = U_{\text{в\ddot{в}}} \frac{R_x}{R_l} = U_{\text{в\ddot{в}}} \frac{x}{l} = kx, \quad (1.1)$$

де $U_{\text{в\ddot{в}}}$ - вхідна напруга (напруга живлення) датчика, R_x - вихідний опір датчика при положенні движка x , R_l - вхідний опір датчика, k - постійний коефіцієнт.

1.1.3.3. Тензометричні датчики

Використовують для вимірювання переміщень – деформації деталей. Принцип дії базується на явищі тензоефекту – зміни величини активного опору провідникового або напівпровідникового матеріалу під дією прикладених до них механічних напружень (рис. 1.3).

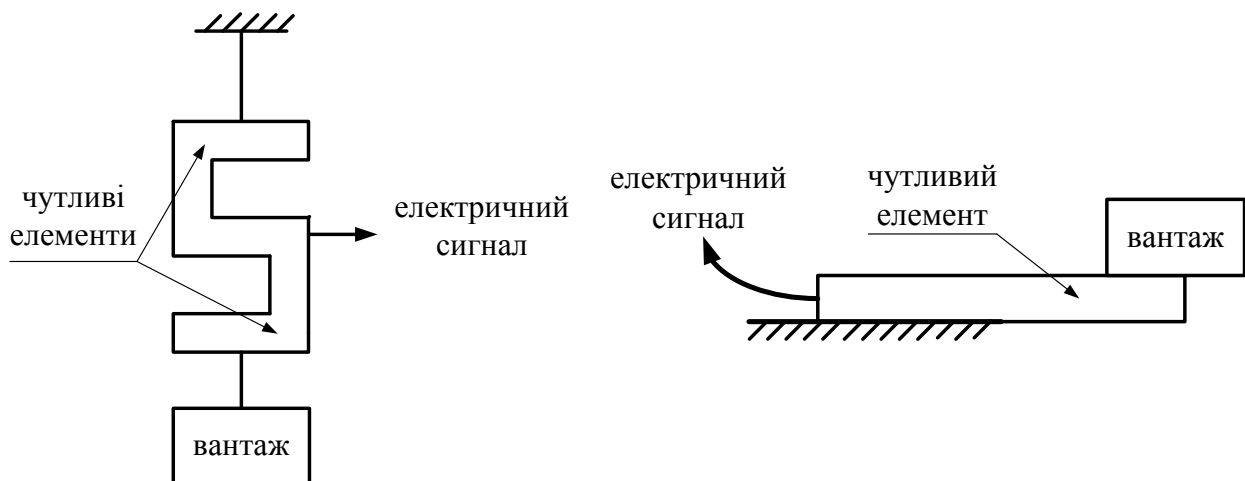


Рис. 1.3. Принцип дії тензометричного датчика

Основна їх характеристика – коефіцієнт тензочутливості – відношення зміни опору до зміни деформації:

$$k = \frac{\Delta R/R}{\Delta l/l}, \quad (1.2)$$

де Δl та ΔR - абсолютна деформація датчика (лінійна) та викликана нею абсолютна зміна його вихідного опору, R - вхідний опір датчика, l - розмір датчика у недеформованому стані. Мостова схема вмикання тензометричного датчика приведена на рис. 1.4 [15].

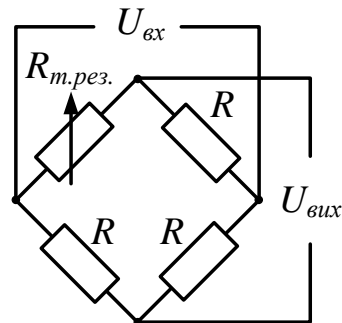


Рис. 1.4. Схема вмикання тензометричного датчика

1.1.3.4. П'єзоелектричні датчики

Дія заснована на явищі п'єзоефекту, суть якого полягає в тому, що під дією прикладених зусиль на гранях деяких кристалів (кварц, титан, турмалін) з'являється електричний заряд (прямий п'єзоефект) (рис. 1.5). При внесенні п'єзоелементу в електричне поле він деформується (обернений п'єзоефект).

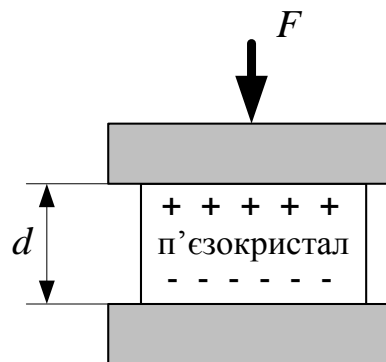


Рис. 1.5. Принцип дії п'єзоелектричного датчика

Вихідний сигнал датчика:

$$U_{\dot{a}\dot{e}\dot{o}} = \frac{Qd}{\epsilon f}, \quad (1.3)$$

де Q_x – кількість зарядів на п'єзовісі (вісь вздовж якої діє сила); f - площа грані кристала перпендикулярної до п'єзовісі; ε - діелектрична проникливість матеріалу кристала; d – товщина пластини кристала.

Якщо сила F діє постійно, то проходить стікання зарядів і вихідна напруга змінюється. Тому п'єзоелементи застосовуються в основному для вимірювання зусиль, що змінюються з частотою $> 15\text{кГц}$. П'єзоелементи використовуються з підсилювачами.

1.1.3.5. Датчик Холла

Такий датчик застосовується для вимірювання напруженості магнітного поля. Практично ж промислові датчики такого роду реалізують на базі напівпровідників. Принцип дії: якщо пластинка напівпровідника помістити у магнітне поле з напруженістю H , а по пластині тече струм величиною I при цьому лінії напруженості електричного поля знаходяться під прямим кутом з напрямком протікання струму, то на бокових гранях цієї пластини виникає напруга (рис. 1.6).

Вихідний сигнал датчика:

$$U_{\text{агв}} = K_H IH, \quad (1.4)$$

де K_H – постійна величина, яка залежить від матеріалу пластини; I – сила струму, який протікає по пластині; H – напруженість магнітного поля.

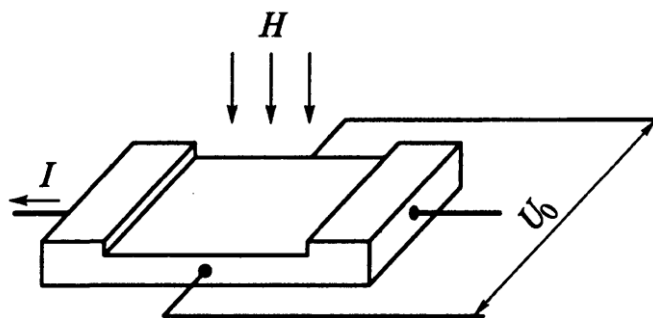


Рис. 1.6. Принцип дії датчика Холла

За допомогою датчиків Холла можна вимірювати кутові і лінійні переміщення (на зірочку ставиться магніт), електричні струми тощо.

1.1.3.6. Ємнісні датчики

Зміна входньої величини викликає зміну ємності конденсатора. Ємність плоского конденсатора:

$$C = \frac{\epsilon_0 \epsilon F}{d}, \quad (1.5)$$

де ϵ_0 – діелектрична проникність вакууму (повітря); ϵ – діелектрична проникність середовища між пластинами; F – площа пластин; d – величина зазору. Варіанти виконання ємнісних датчиків показані на рис. 1.7.

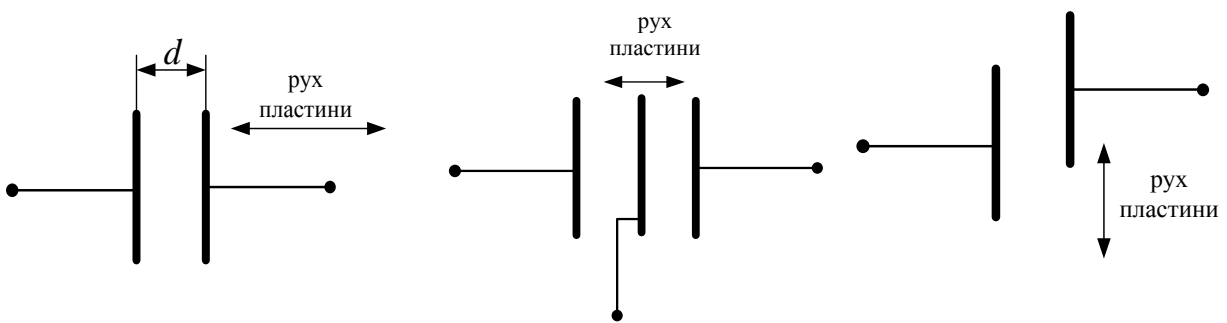


Рис. 1.7. Варіанти виконання ємнісних датчиків

Часто такі типи датчиків використовують для вимірювання прискорення (акселерометри), вібрації, нахилу основи, на якій вони знаходяться.

Ємнісні датчики також використовуються й для виміру вологості. У цьому випадку датчик являє собою конденсатор, що складається з діелектричної плівки, на обидві сторони якої шляхом напилювання наносяться електроди із золота. Ємність такого датчика виявляється приблизно прямо пропорційною вологості навколишнього середовища.

1.1.3.7. Оптичні датчики

Оптоелектроніка сполучає в собі оптичні й електронні методи вимірів. На основі оптоелектронних перетворювачів створені датчики тиску, сили, переміщення, швидкості, акустичних параметрів, напруженості електричного й магнітного полів. Для параметричних оптичних датчиків необхідно використовувати джерело світлового випромінювання. Найпоширенішими джерелами світлового випромінювання, які використовуються в мехатронних системах, є світлодіоди.

Приймачі випромінювання можна поділити на дві групи: інтегральні й селективні. До *інтегрального* відносяться приймачі випромінювання, що базуються на перетворенні енергії випромінювання в сигнал незалежно від довжини хвилі цього випромінювання. До *селективного* відносяться фотоелектричні перетворювачі, що реагують на певну довжину хвилі випромінювання.

Фоторезистори являють собою напівпровідникову пластинку з контактами, що при освітленні зменшує свій опір. Фотодіоди й фототранзистори відносяться до напівпровідникових приймачів випромінювання. Фототранзистор являє собою той же фотодіод, який має підсилювач струму.

Принцип роботи оптичного датчика та його вихідний сигнал показано на рис. 1.8.

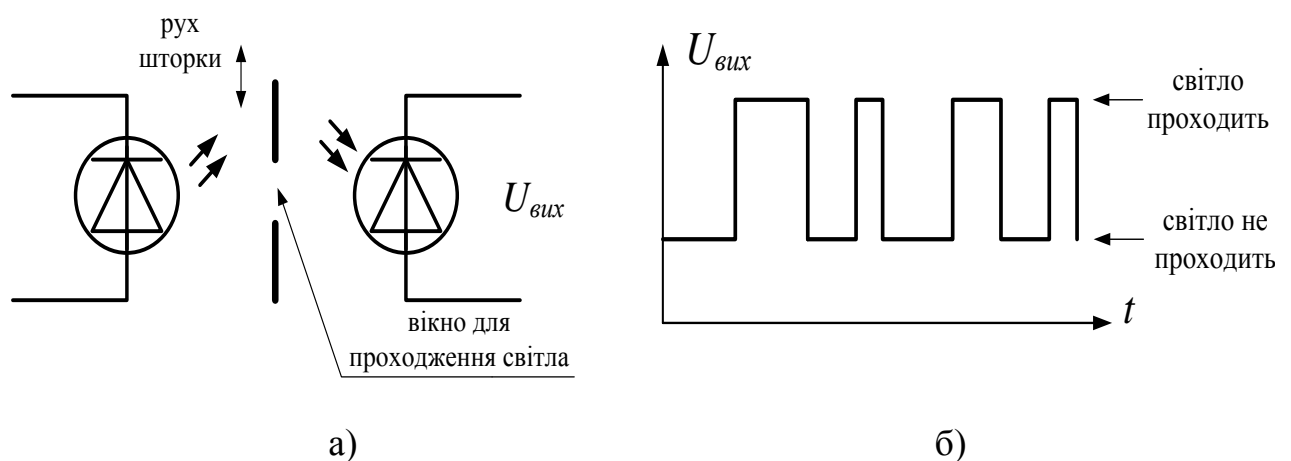


Рис. 1.8. Принцип дії оптичного датчика (а) та його вихідний сигнал (б)

Одним з найпоширеніших пристроїв у мехатронних системах є енкодер – сенсор, який призначений для вимірювання кутових та лінійних переміщень, швидкостей та прискорень. Принцип роботи оптичного енкодера заснований на перетинанні променя оптопари (світлодіод та фототранзистор) з розміченим диском, установленим на валу (рис. 1.9). Залежно від частоти розмітки диска визначається точність виміру переміщення.

Існує кілька різновидів енкодерів, найбільш використовуваних з яких інкрементний (імпульсний), де відбувається послідовний рахунок міток переміщення й абсолютний, коли для кожного положення вала існує індивідуальний бінарний (двійковий) код. Інкрементний енкодер призначений для формування імпульсів, які зчитуються вторинним пристроєм, що дозволяє визначити напрямок руху або кутового зсуву механізму. Звичайно інкрементні енкодери виготовляють із трьома імпульсними виходами. Два з них визначають швидкість і напрямок руху (канали А і В). Послідовності імпульсів цих виходів зсунуті один відносно одного, що реалізовано просторовим зміщенням оптичного диска й розташуванням оптичних пар „випромінювач-приймач”. Третя послідовність імпульсів служить для визначення позиції (канал Z). Цей канал формує імпульс нульової оцінки на один оберт, що дозволяє корегувати помилки в межах кожного оберту. Підрахунком імпульсів за один оберт від нульової точки можна визначати поточне абсолютне положення вала енкодера.

На відміну від інкрементного енкодера, лічильник імпульсів не потрібний, тому що кут повороту завжди відомий. Абсолютний енкодер формує сигнал як під час обертання, так й у режимі спокою. Диск абсолютного енкодера відрізняється від диска покрокового енкодера, тому що має кілька концентричних доріжок. Кожною доріжкою формується унікальний двійковий код для конкретної позиції вала (сектора) (рис. 1.9).

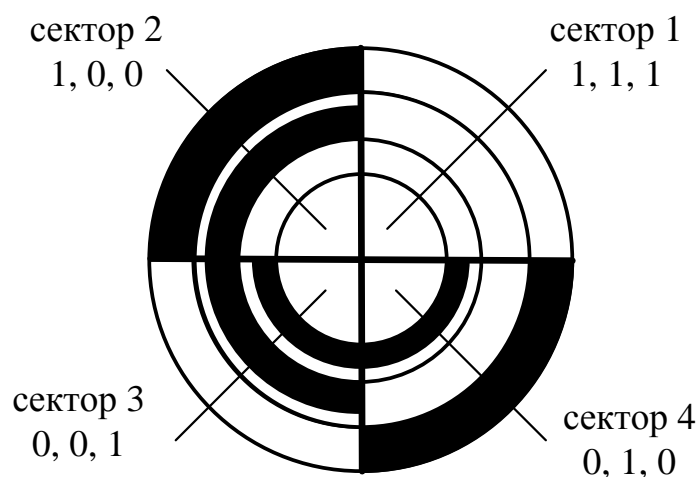


Рис. 1.9. Диск абсолютного енкодера

Абсолютний енкодер не втрачає значення сигналу при втраті живлення й не вимагає повернення в початкову позицію. Сигнал абсолютного енкодера не може бути пошкоджений перешкодами і для нього не потрібна точна установка вала. Крім того, навіть якщо кодований сигнал не може бути прочитаний енкодером якщо, наприклад, вал обертається занадто швидко, правильний кут обертання буде зареєстрований, коли швидкість обертання зменшиться. Абсолютний енкодер стійкий до вібрацій.

1.1.3.8. Електромагнітні датчики

Принцип дії заснований на зміні індуктивності (L) магнітної системи під дією вхідної величини (переміщення, деформація, рівень, тиск). За видом перетворення електромагнітні датчики поділяються на: індуктивні; трансформаторні; магнітопружні; індукційні.

Індуктивні датчики – перетворюють зміну керованої величини в зміну індуктивного опору обмотки. Вихідний сигнал датчика:

$$U_{\text{вх}} \approx K_{\delta} \delta, \quad (1.6)$$

де $U_{\text{вх}}$ - вхідна напруга; K_{δ} - постійний коефіцієнт; $U_{\text{вих}}$ - вихідна напруга.

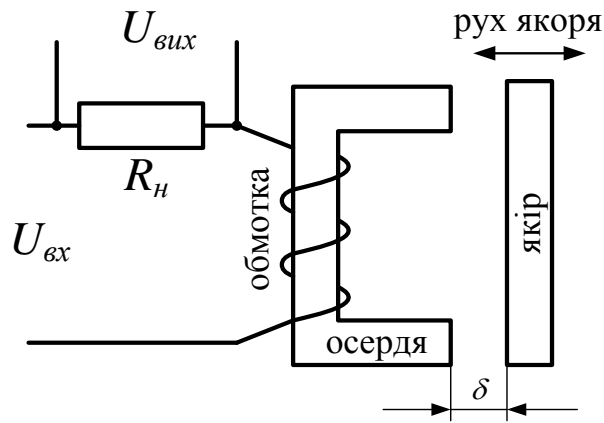


Рис. 1.10. Принцип дії індуктивного датчика

Датчики з аналоговим виходом широко застосовуються в області виміру й контролю: вимір відстані переміщення металевих предметів; контроль товщини, ширини листового або стрічкового матеріалу; вимір кута повороту кулачків і шестерень; контроль і вимір биття валів обертання; у системах контролю й регулювання натягу транспортних стрічок, тросів; селективний відбір виробів різних розмірів і з різних металів (рис. 1.11).

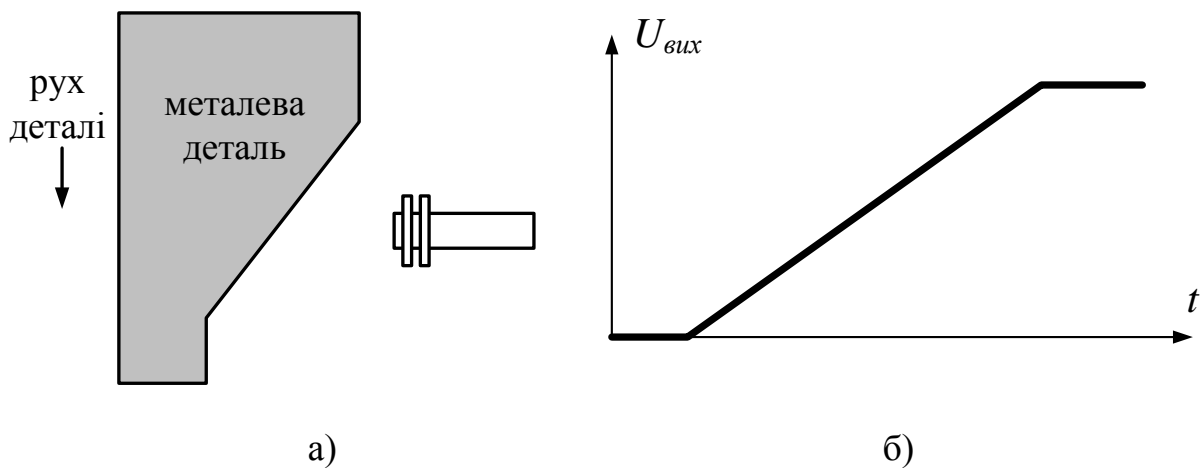


Рис. 1.11. Робота індуктивного датчика (а) та сигнал із його виходу (б)

Трансформаторні датчики

В основі роботи лежить залежність зміни взаємної індуктивності обмоток (первинної та вторинної) при переміщенні однієї відносно іншої або при переміщенні якоря датчика (рис. 1.12).

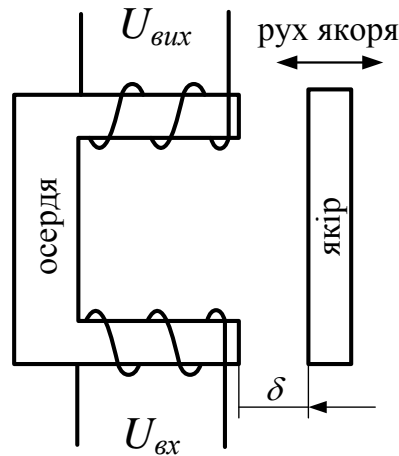


Рис. 1.12. Трансформаторний датчик

Для виміру кутових переміщень виконують у вигляді електромашин – обертові трансформатори (сельсини – у яких первинна обмотка розташовується на статорі, а вторинна – на роторі).

Магнітопружні датчики

Використовують зміну магнітної проникливості феромагнітних тіл, які підлягають дії сил або температур. Чутливість $k = \frac{\Delta\mu/\mu}{\Delta l/l}$ (зміна магнітної проникливості від зміни лінійних розмірів). Недоліком даних датчиків є залежність властивостей від температури та не лінійність характеристики магнітної проникливості від прикладеної сили.

Індукційні датчики

Їх дія заснована на явищі електромагнітної індукції – наведення е.р.с. в електроконтурі, в якому змінюється магнітний потік. До них відносять тахогенератори постійного та змінного струму.

1.1.3.9. Ультразвукові датчики

Ультразвукові датчики можна встановлювати горизонтально, вертикально або під кутом до об'єкта. Принцип дії заснований на передачі ультразвукових імпульсів і визначенні часу їх відбивання і надходження назад до датчика. У якості чутливого елементу датчика виступає

п'єзоелектричний кристал. Він же генерує ультразвук. Час між випромінюванням ультразвукового імпульсу та прийманням відбитого імпульсу є основою для розрахунку відстані до об'єкта (рис. 1.13). Повний контроль за процесом виміру здійснюється з мікропроцесором.

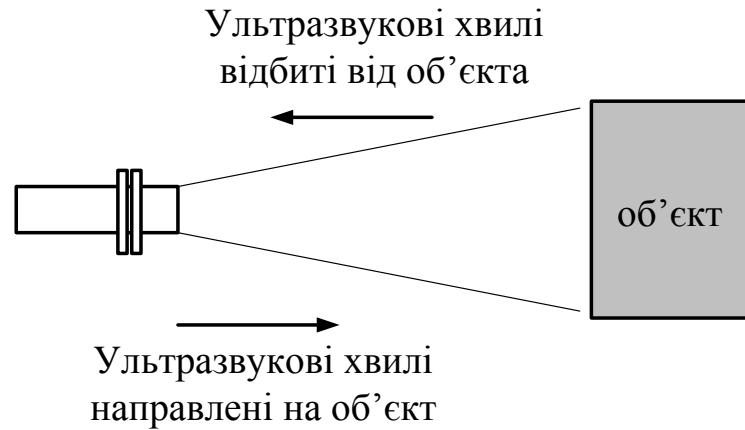


Рис. 1.13. Принцип дії ультразвукового датчика

1.1.3.10. Датчики температури

Бувають з механічним вихідним сигналом (*тепломеханічні*) та з електричним вихідним сигналом (*теплоелектричні*).

Тепломеханічні (термобіметалеві). Використовуються як елементи мехатронних систем, які перетворюють зміну фактичного значення температури, що вимірюється, в переміщення. Термобіметалевий датчик являє собою вузьку пластину з біметалу (зварені між собою дві пластинки металу з різними коефіцієнтами розширення), один кінець якої закріплений нерухомо. Термобіметалеві датчики використовуються для вимірювання температури в діапазоні $-60 +450$ °С. Похибка 1...5%.

Теплоелектричні датчики (металеві та напівпровідникові терморезистори, термопари, напівпровідникові діоди та тріоди). Перетворюють фактичне значення температури в зміну електричного опору, напруги, струму.

Металеві терморезистори виготовляють з чистих металів (платина, мідь, нікель, залізо, вольфрам). Похибка вимірювання металевих терморезисторів складає менше 0,5...1%.

Напівпровідникові терморезистори. Діапазон температур 203...523 °К
Чутливість – в 5...30 раз вища металевих терморезисторів. Недолік – нелінійна характеристика, погана взаємозамінність (розкид номінальних параметрів).

Термопари – спай з двох різнорідних металів або напівпровідників. При $t_1 > t_2$ виникає термоелектрорушійна сила. Діапазон температур 373...3000 °К. Чутливість 0,01...0,07 мВ на 1 °С. Похибка до 5%. Недоліки: не терплять проміжних з'єднань (необхідно застосовувати схеми термокомпенсації), мають невисоку чутливість.

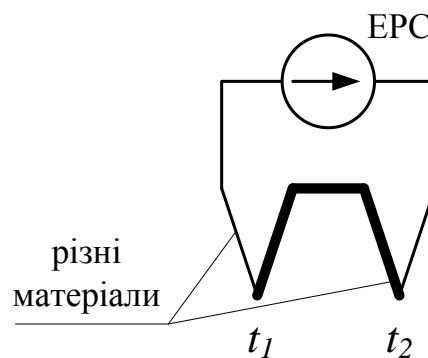


Рис. 1.14. Принцип дії термопари

1.1.3.11. Датчики вологості

Гігрорезистори (гігристи) – резистори виготовлені з матеріалів (хлор-літій, калій-натрій, напівпровідників), опір яких різко падає при збільшенні вологості. При підвищенні вологості з 35...40% до 95...100% опір зменшується від 10^7 до 10^2 Ом. Похибка не перевищує 2%.

1.1.4. Аналогова та цифрова (дискретна) форма представлення даних

Зупинимось докладно на двох загальноприйнятих в сучасній вимірювальній техніці формах подання вимірюваних величин: аналоговій та цифровій [14].

Термін „аналоговий” схожий за змістом терміну „подібний”. Він означає, що одна фізична величина (наприклад, вимірювана) виражається за

допомогою іншої величини. Якщо перша величина приймає безперервний ряд значень у якомусь діапазоні, то при аналоговому представленні даних друга величина приймає безперервний ряд значень у відповідному діапазоні. Це означає, що вихідна величина є моделлю, аналогом вхідної величини.

Поряд з аналоговою застосовується й інша форма подання вимірювальної інформації – дискретна. Тут вимірюваний сигнал може приймати лише деяке кінцеве число значень, причому ці значення можуть встановлюватись як у будь-який момент часу (неперервний час), так і у певні моменти часу (дискретний час). Отже можна говорити про дискретизацію часу та дискретизацію величини сигналу (рис. 1.15).

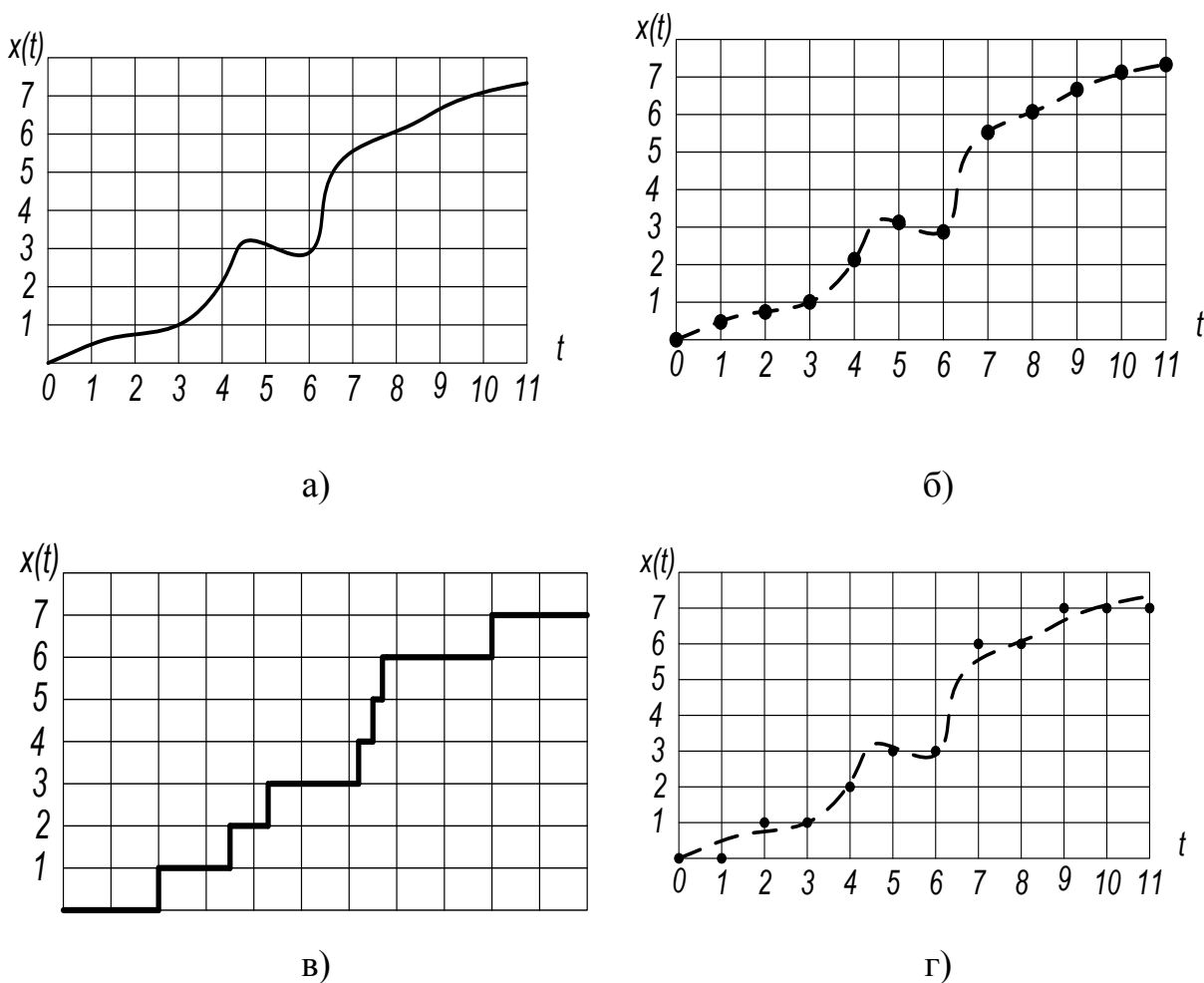


Рис. 1.15. Форми сигналів: а) аналоговий сигнал $x(t)$ в функції неперервного часу t ; б) аналоговий сигнал $x(t)$ в функції дискретного часу t ; в) дискретний сигнал $x(t)$ в функції неперервного часу t ; г) дискретний сигнал $x(t)$ в функції дискретного часу t

Для перетворення аналогової форми сигналу у цифрову використовують аналогово-цифрові перетворювачі (АЦП), навпаки: з цифрової у аналогову цифро-аналогові перетворювачі (ЦАП).

1.1.5. „Інтелектуальні” датчики

У цей час продається безліч датчиків з вбудованим мікропроцесором. Такі датчики прийнято називати „інтелектуальними” [16]. Обробка вимірів безпосередньо в самому датчику дозволяє поліпшити його характеристики. Загальна схема таких датчиків показана на рис. 1.16.

Виходи інтелектуальних датчиків можуть бути аналоговими й цифровими. Аналогові системи використовують рівні сили струму 4-20 мА, коли нульове значення сигналу передається рівнем 4 мА, щоб можна було відрізнити обрив кабелю від нульового значення сигналу. Стандартний робочий діапазон: 1-5 В або, для кращої завадостійкості на довгих відстанях: 4-20 мА. Надалі масштаби впровадження мікропроцесорної обробки сигналів в датчиках будуть зростати внаслідок високої ефективності й конкурентоздатності таких систем.

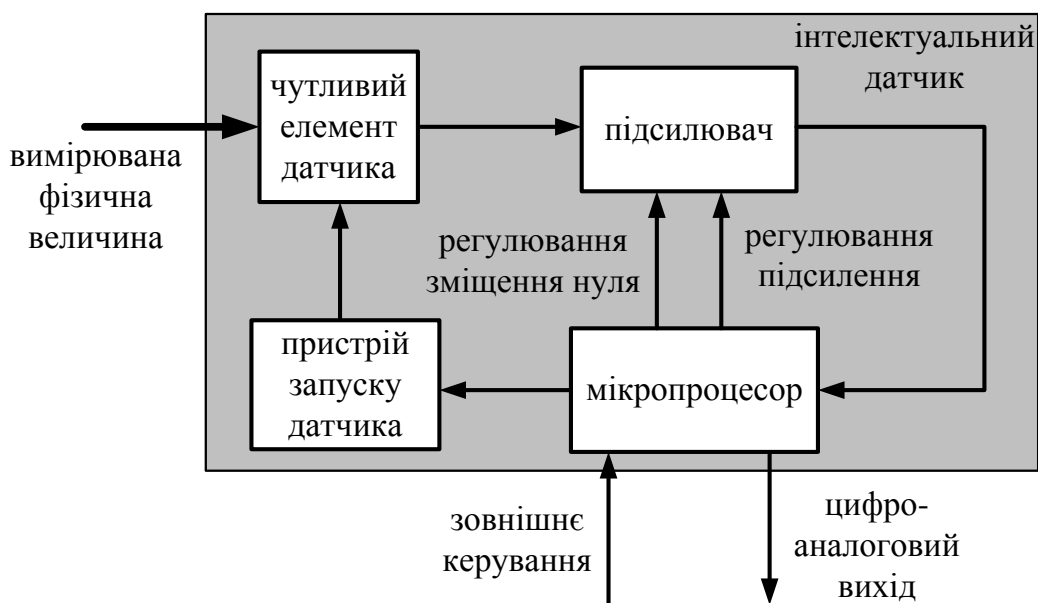


Рис. 1.16. Архітектура „інтелектуальних” датчиків

Інтелектуальні датчики, звичайно, мають велику кількість варіантів виконання, розрахованих на різні властивості вимірюваної величини та навколишнього середовища:

- ▶ варіанти корпусу сенсора під різні тиски, температури, впливи й перешкоди;
- ▶ варіанти матеріалів арматури, що контактують із вимірюваним середовищем, під звичайне, хімічно агресивне, абразивне й інші середовища;
- ▶ варіанти виконання сенсора під звичайне, гігієнічне, вибухонебезпечне середовище;
- ▶ варіанти з'єднання сенсора з конструкцією об'єкта виміру (фланцеве, різьбове тощо).
- ▶ варіанти живлення приладу за наявності блоку живлення в ньому або живленню його від стороннього джерела через польову мережу;
- ▶ варіанти вихідних сигналів перетворювача по числу, по параметрах, по комунікаційних можливостях зв'язку з різними польовими мережами.

Дуже важливо, що більшість виробників комплектують датчики з комбінації різних варіантів сенсорів одного методу виміру з різними варіантами перетворювачів (електронного обладнання інтелектуального датчика), розрахованих на роботу з даною серією сенсорів. Завдяки цьому вдається найбільш точно й повно задовольняти окремим конкретним вимогам до приладів. Слід мати на увазі, що подібна технічно раціональна гнучкість побудови датчиків, у той же час, не дозволяє, у ряді випадків, дати оцінку вартості приладу без детального аналізу обраних варіантів складових його блоків.

Інтелектуальні датчики є багатофункціональними приладами, для яких тільки традиційно зберігається найменування „датчик”, а за виконуваними функціями вони все більше наближаються до симбіозу датчика й контролера. Тенденція їх розвитку, пов'язана із усіма можливостями вбудованих у них мікропроцесорів, полягає в передачі їм від контролерів все більшого числа

найпростіших типових функцій контролю й керування. Крім того, сучасні інтелектуальні датчики ширше використовують можливості свого мікропроцесорного перетворювача для вдосконалювання процесу вимірювання: підвищення точності, збільшення надійності, вибору діапазону виміру, виключення помилкових вихідних даних, розширення функцій дистанційного керування роботою сенсора.

Інформаційні функції. Датчики можуть зберігати дані, що визначають властивості, характеристики, параметри даного конкретного приладу: його тип, заводський номер, технічні показники, можливі діапазони виміру, установлену шкалу, задані параметри налаштування, версію програмного забезпечення, архів наступної перевірки датчика тощо. Ці дані, за запитом користувача, можна зчитати. Крім того, датчики можуть мати архів поточних вимірюваних значень та значень, які обчислюються ними за заданий інтервал часу.

Функції конфігурування. Дистанційне формування або модифікація користувачем основних налаштувальних параметрів датчика: установка нуля приладу, вибір заданого діапазону виміру, фільтрація поточних значень, вибір найменування одиниць виміру, у яких датчик повинен видавати інформацію тощо.

Функції форматування. Автоматичний аналіз змін вимірюваної величини й поточного стану середовища виміру: визначення виходів значень вимірюваної величини за задані межі, видача різних повідомлень про зміни значень вимірюваної величини, перевірка знаходження в припустимих діапазонах параметрів вимірюваного середовища. Усі ці функції дистанційно настроюються користувачем.

Функції самодіагностики. У процесі роботи датчики виконують аналіз своєї роботи: при виникненні різних збоїв, порушень і несправностей фіксують їхнє місце виникнення й причину, визначають вихід похибки приладу за паспортну норму, аналізують роботу бази даних датчика,

розглядають правильність врахування факторів, які корегують вихідні покази датчика.

Функції перетворення. Датчик перетворює електричний сигнал на виході чутливого елемента (звичайно, низьковольтний аналоговий, частотний або імпульсний сигнал) у значення заданої одиниці виміру, при цьому він виконує корекцію вихідного значення по супутніх поточних показниках стану вимірюваного середовища (наприклад, по її температурі або тиску), випадку, якщо показання датчика залежать і від них. У приладі проводяться необхідні перетворення вимірювальної інформації: посилення сигналів сенсора, стандартизація діапазонів вихідних аналогових сигналів, лінеаризація й фільтрація вимірюваних значень, розрахунки вихідних значень по заданих алгоритмах, аналого-цифрове перетворення значень вимірюваної величини.

1.2. Привод мехатронних систем

1.2.1. Вимоги до приводів

Сучасні мехатронні системи повинні мати приводи, до яких пред'являються високі й досить суперечливі вимоги. Структура приводів повинна забезпечувати можливість автономного й координованого (погодженого) регулювання переміщень, швидкостей, сил (моментів) за умови забезпечення високої точності позиціонування при відсутності перерегулювання, високого коефіцієнта форсування по моменту, максимальної швидкодії, плавності переміщення робочого органу мехатронної системи при змінних навантаженнях і моментах інерції, надійної фіксації в точках позиціонування, відсутності перехідних режимів при перемиканні структури привода, а також мінімальність маси й габаритів електромеханічних модулів при високій надійності всіх елементів.

Особливі умови експлуатації мехатронних систем накладають істотні обмеження на вибір типу двигуна привода. Застосування крокового

електропривода малоефективне при потужностях, що перевищують сотні Ват. Електроприводи на базі лінійних двигунів, поряд з очевидними перевагами забезпечення лінійного переміщення виконавчих пристроїв без механічних передач, не забезпечують необхідної точності й швидкодії

1.2.2. Гідравлічні і пневматичні приводи

Основним елементом виконавчих органів, робота котрих заснована на використанні рідини, що знаходиться під тиском, чи стиснутого повітря, є силовий гідро (пневмо) циліндр. У циліндрі під дією напору повітря (рідини), яке надходить від джерела, у котрому воно утримується під тиском, відбувається переміщення поршня (рис. 1.17) [7].

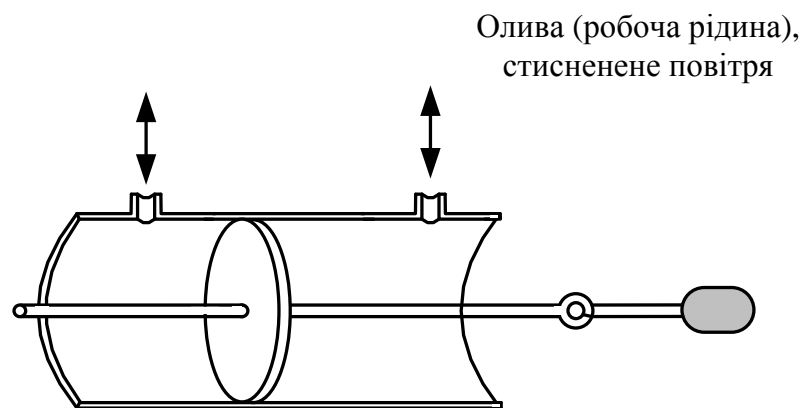


Рис. 1.17. Силовий гідро (пневмо) циліндр

Не дивлячись на свої розміри, гідравлічні й пневматичні приводи дозволяють розвинути значні зусилля. Однак вони вимагають створення спеціальних джерел, з котрих повинна надходити рідина високого тиску чи стиснутий газ. При використанні гідроприводу можуть виникнути проблеми, пов'язані з захистом оточуючого середовища від забруднень, витіканні гідравлічної рідини. При використанні пневмоприводу такої проблеми не виникає. Однак, при стискуванні повітря виникають труднощі, пов'язані з керуванням.

Застосування пневматичних приводів у мехатронних системах пояснюється їхньою дешевиною, простотою й відповідно надійністю. Правда,

ці приводи погано керовані. Пневматичні приводи застосовують тільки в роботах невеликої вантажопідйомності - до 10 кг, рідше 20 кг. Гідравлічні приводи більш складні й дороги в порівнянні із пневматичними й електричними. Однак при потужності 500-1000 Вт і вище вони мають найкращі массогабаритні характеристики й тому є основним типом привода для важких і надважких мехатронних систем. Гідравлічні приводи добре керуються, тому вони знайшли також застосування в мехатронних системах середньої вантажопідйомності, для яких потрібні високоякісні динамічні характеристики.

1.2.3. Електропривод. Основні поняття та класифікація

Електроприводом називають електромеханічну систему, що складається з електродвигунного, перетворювального, механічного передавального і керуючого пристроїв і призначена для приведення в рух виконавчого органу робочої машини та керування цим рухом [17].

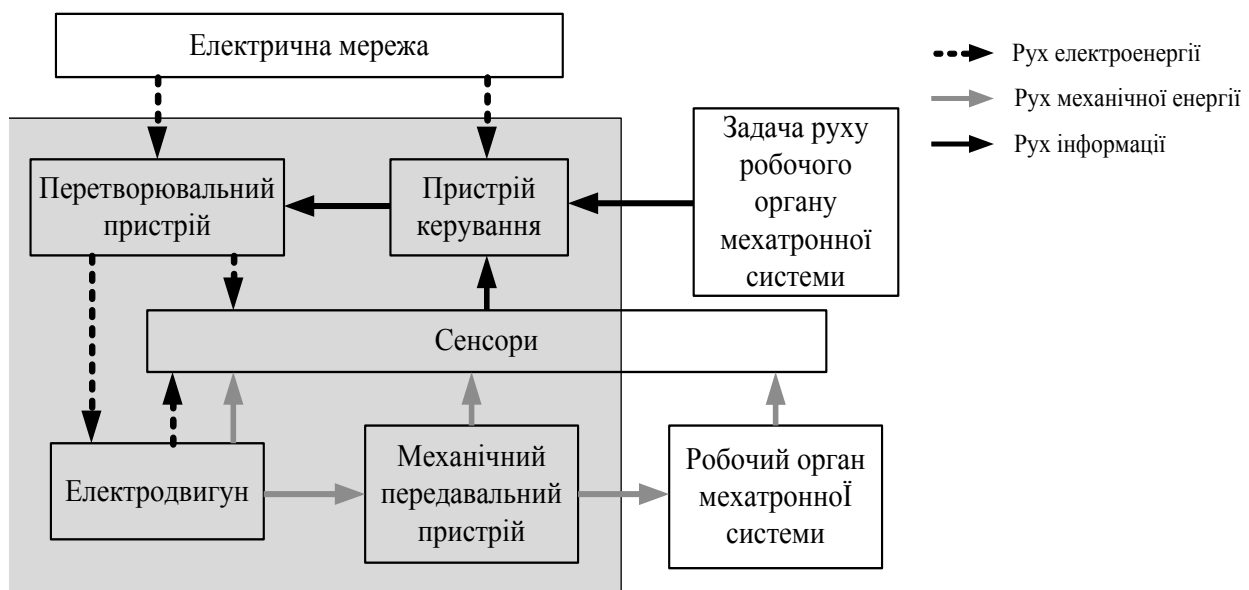


Рис. 1.18. Структура електроприводу у складі мехатронних систем

До переваг електроприводу можна віднести те, що для нього джерелом енергії слугує мережа постійного або змінного струму. Відсутня необхідність у використанні трубопроводів, а також проблема забруднення навколишнього середовища.

Електродвигуни можна класифікувати таким чином (рис. 1.19).



Рис. 1.19. Класифікація електродвигунів

1.2.4. Електропривод постійного струму

1.2.4.1. Колекторний електропривод

Основним елементом електропривода мехатронної системи зазвичай є електродвигун постійного струму, на виході котрого крутний момент пропорційний силі струму на вході двигуна. У такому електродвигуні подача струму до ротора здійснюється через щітки, котрі доволі легко можна пошкодити і тому вони вимагають необхідної уваги при обслуговуванні [18].

Принцип дії двигуна постійного струму ілюструє рис. 1.20, він полягає у взаємодії магнітних полів полюсів статора і провідників ротора, по яких протікає струм. Якщо струм у провіднику йде від нас (на рис. 1.20 провідник по якому струм йде від нас позначений колом із хрестиком), то, за правилом гвинта, магнітні силові лінії навколо провідника спрямовані за годинниковою стрілкою. У результаті додавання магнітних полів полюсів і провідника підсилиться магнітне поле праворуч від провідника ротора й послабиться ліворуч від нього. До провідника буде прикладена сила, яка виштовхує його у бік ослабленого магнітного поля, тобто вліво. Аналогічні процеси відбуваються з провідником, який знаходиться над південним полюсом

статора (цей провідник показано колом із точкою – струм по ньому йде до нас).

На рис. 1.21 показано схему вмикання електричного двигуна постійного струму із незалежним збудженням. Електромагнітний момент, що виникає у двигуні виражається формулою:

$$M = K' \Phi \cdot I_{\text{ротор}} = K'' I_{\text{збудження}} \cdot I_{\text{ротор}}, \quad (1.7)$$

де K' – коефіцієнт, що залежить від конструкції ротора двигуна; Φ – магнітний потік двигуна; $I_{\text{ротор}}$ – струм обмотки ротора; $I_{\text{збудження}}$ – струм кола збудження двигуна; K'' – коефіцієнт, що залежить від конструкції ротора та статора двигуна.

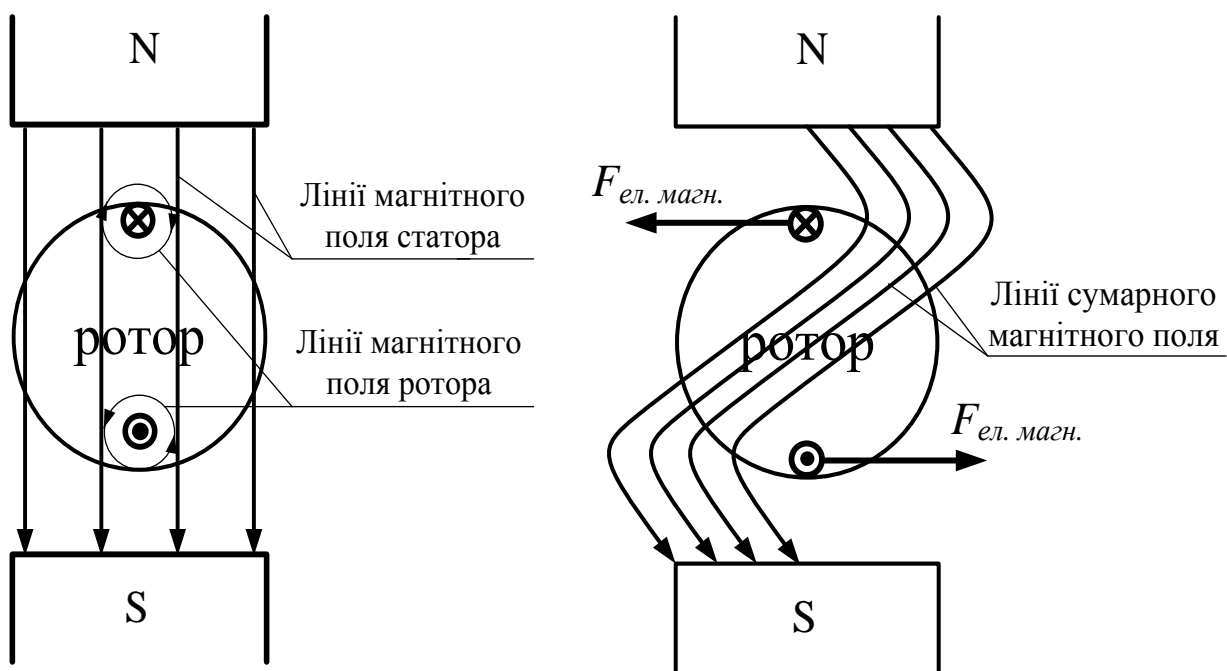


Рис. 1.20. Принцип виникнення електромагнітного моменту у двигуні постійного струму

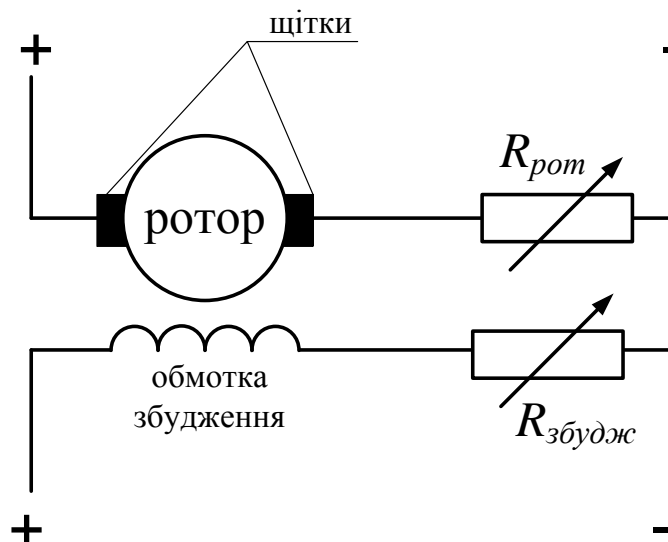
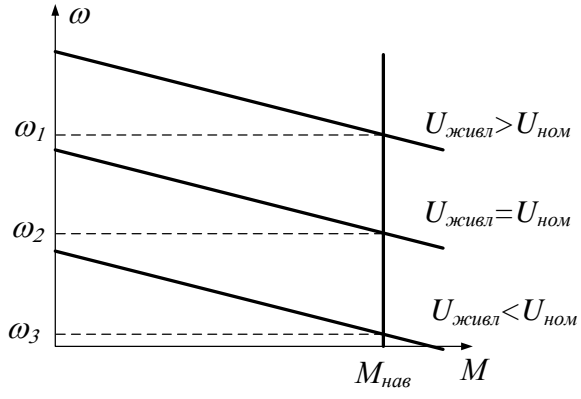
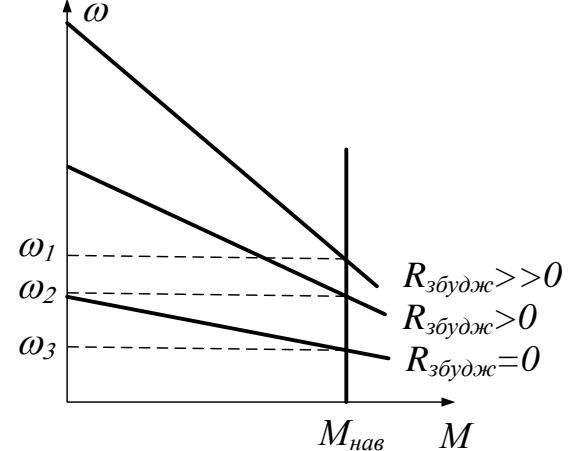


Рис. 1.21. Електрична схема вмикання двигуна постійного струму незалежного збудження

Розглянемо методи регулювання двигуна постійного струму (табл. 1.2).
 Регулювати двигун постійного струму можна зміною: 1) напруги живлення;
 2) опору роторного кола; 3) опору кола збудження.

Таблиця 1.2. Способи регулювання двигуна постійного струму послідовного збудження

Спосіб	Графіки механічних характеристик	Коментар
1	2	3
1. Зміна опору роторного кола		Зі збільшенням опору роторного кола швидкість двигуна падає.

1	2	3
2. Зміна напруги живлення		Зі збільшенням напруги живлення двигуна його швидкість збільшується, і навпаки, зі зменшенням напруги живлення його швидкість зменшується.
3. Зміна струму у колі збудження		Зі збільшенням опору кола збудження швидкість двигуна збільшується. При обриві кола збудження швидкість зростає до нескінченності і двигун іде „вразнос”.

При регулюванні двигуна постійного струму останній має гарні показники: діапазон регулювання швидкості до 1:100000, перевантажувальна здатність по моменту до 8...10. Разом з тим, електроприводи постійного струму мають один істотний недолік – наявність щіткового колектора. Великий обсяг планово-запобіжних робіт з обслуговування і контролю двигунів навіть у приміщеннях з нормальним середовищем, а особливо у

зволожених, запилених і тим більше при наявності жорстких випромінювань приводить до зниження економічного ефекту при мехатронізації виробництва через істотне зростання витрат на ремонт і заміну електродвигунів постійного струму.

1.2.4.2. Кроковий електропривод

Недоліки двигуна постійного струму полягають перша за все у наявності колектора. Тому активно ведуться розробки безколекторних, наприклад, крокових двигунів. Принцип дії такого двигуна ілюструє рис. 1.22.

Двополюсний ротор із видовженими виступами поміщений у чотириполюсний статор. Одна пара полюсів виконана з постійних магнітів, на іншій – намотана обмотка керування. Поки струму в обмотках керування немає, ротор орієнтується уздовж постійних магнітів і утримується з певним зусиллям, яке визначається магнітним потоком полюсів [19].

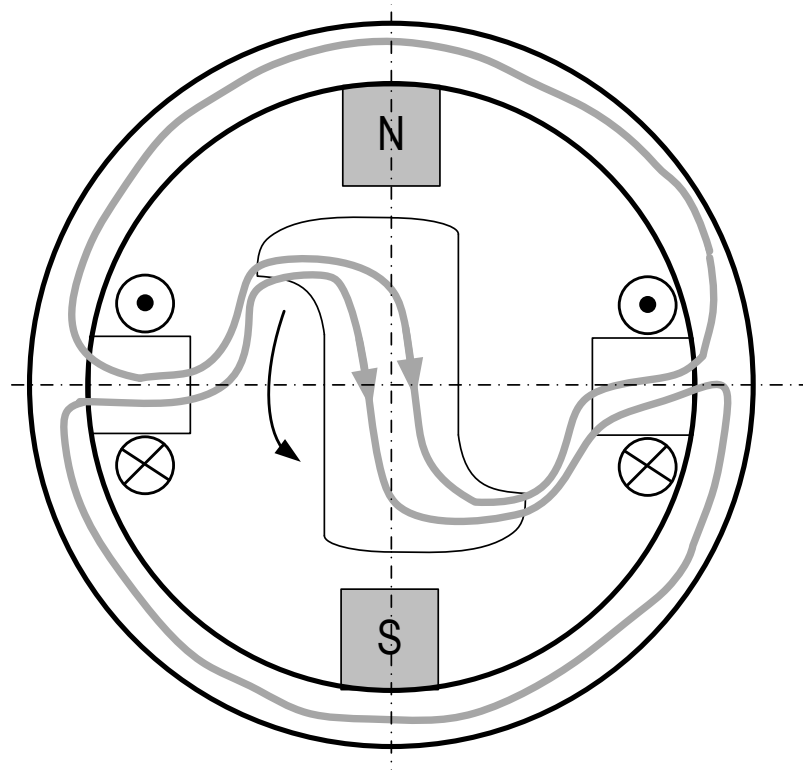


Рис. 1.22. Принцип дії крокового двигуна

При подачі постійної напруги на обмотку керування виникає магнітний потік приблизно вдвічі більший, ніж потік постійних магнітів. Під дією електромагнітного зусилля, створюваного цим потоком, ротор повертається, долаючи момент навантаження і момент, що розвивається постійними магнітами, прагнучи зайняти положення співвісне з полюсами керуючої обмотки. Поворот відбувається у бік видовжених виступів, тому що магнітний опір між статором і ротором у цьому напрямку менший, ніж у зворотному. Наступний керуючий імпульс відключає напругу з обмотки керування й ротор повертається під дією потоку постійних магнітів у бік виступів. Перевагою крокових двигунів з постійними магнітами є простота конструкції й схеми керування. Для фіксації ротора при знеструмленій обмотці керування не потрібне споживання енергії, кут повороту зберігає своє значення й при відсутності живлення. Двигуни цього типу відпрацьовують імпульси із частотою до 200-300 Гц. Їхній недолік – низький ККД.

Характер руху ротора крокового двигуна визначається частотою й характером зміни керуючих імпульсів. Залежно від цього розрізняють наступні режими роботи крокових двигунів: статичний, квазістатичний, перехідний. Статичний режим – це режим, при якому по обмотках статора протікає постійний струм, що створює нерухоме в просторі магнітне поле, а ротор не обертається. Під дією навантаження ротор лише відхиляється від положення рівноваги на деякий кут. Такий режим використовується для фіксації робочого органу мехатронної системи у деякому положенні (наприклад, фіксація руки робота, яка триває вантаж).

Квазістатичний режим – це режим відпрацьовування одиничних кроків, при якому перехідні процеси від попереднього такту комутації повністю загасли й швидкість ротора на початку наступного кроку дорівнює нулю. Він використовується в механізмах, у яких потрібна фіксація ротора після кожного кроку. Гранична частота керуючих імпульсів, при якій ще витримується квазістатичний режим, визначається часом протікання

електромагнітних і особливо електромеханічних перехідних процесів, тобто часом коливань ротора. Підвищити граничну частоту руху ротора квазістатичного режиму можна шляхом збільшення числа обмоток статора або числа тактів комутації. У всіх цих випадках зменшується кут переміщення й кінетична енергія ротора, що зменшує його коливання у кінці кроку.

Перехідний режим - це основний експлуатаційний режим роботи крокового двигуна. Він містить у собі пуск, реверс, гальмування, перехід з однієї керуючої частоти на іншу. Фізичні процеси в перехідних режимах визначаються як параметрами двигуна і його навантаженням, так і початковими умовами, при яких починається перехідний процес. Основна вимога до крокового двигуна в перехідних режимах полягає у відсутності втрати кроку, тобто збереження синхронізму при будь-якому характері зміни керуючих імпульсів. Пуск крокового двигуна здійснюється з нерухомого положення ротора шляхом стрибкоподібного збільшення частоти керуючих імпульсів (від нуля до робочої). При цьому ротор спочатку відстає від поля, потім, прискорюючись, досягає частоти обертання поля, випереджає його й внаслідок від'ємного синхронізуючого моменту знову сповільнює свій рух. Внаслідок демпфування коливань швидкості обертання ротора вони швидко загасають і потім настає усталений режим.

Максимальна частота керуючих імпульсів, при якій можливий пуск без втрати кроку, росте зі збільшенням максимального синхронізуючого моменту, зменшенням кроку, зниженням постійної часу обмоток, навантаження й моменту інерції. Для сучасних крокових двигунів вона дорівнює 100-1000 Гц. Гальмування крокового двигуна здійснюється стрибкоподібним зниженням частоти керуючих імпульсів від робочого значення до нуля. Реверс крокового двигуна проводиться шляхом зміни послідовності комутації струмів в обмотках, що приводить до зміни напрямку обертання магнітного поля на протилежний. Гранична частота керуючих імпульсів, при якій реалізується реверс без втрати кроку, завжди

менше частоти при якій здійснюється пуск і становить 20-50% від цієї частоти.

1.2.5. Електропривод змінного струму

У сучасному автоматизованому виробництві широке застосування знаходять мехатронні системи на базі регульованих електроприводів змінного струму. Асинхронні двигуни (АД) мають ряд переваг перед двигунами постійного струму: більш високу надійність, простоту експлуатації, кращі масогабаритні показники, низьку вартість тощо.

Кутова швидкість АД виражається такою формулою:

$$\omega = \frac{2\pi f}{p}(1-s), \quad (1.8)$$

де f - частота джерела живлення АД, Гц; p - число пар полюсів, s – ковзання двигуна.

Таким чином, швидкість обертання асинхронного двигуна можна регулювати зміною параметрів: частотою струму, числом пар полюсів, опором роторного і статорного кола та величиною напруги на статорі.

1.2.5.1. Частотне скалярне керування асинхронним двигуном

Частотне керування асинхронного двигуна має найкращі показники регулювання швидкості порівняно з іншими способами. Єдиним стримуючим фактором тут є висока, поки що, вартість перетворювача частоти. Цей спосіб забезпечує плавне регулювання швидкості у широкому діапазоні (вниз і ввєрх від номінальної швидкості), а одержані механічні характеристики мають високу жорсткість. Частотний спосіб має важливу властивість - сталість ковзання при регулюванні швидкості. За цієї обставини втрати потужності незначні, тобто цей спосіб є економічним. Типи робочих машин і відповідні їм механічні характеристики АД, що формує ПЧ, приведені у табл. 1.3.

Таблиця 1.3. Робочі машини та механічні характеристики АД

Тип робочої машини мехатронної системи	Механічна характеристика	Коментар
підйомники, конвеєри, крани, маніпулятори		<p>Момент опору робочих машин не залежить від швидкості. Закон зміни напруги живлення АД при зміні частоти $U = fB^*$</p>
вентилятори, насоси, гребні гвинти		<p>Момент опору робочих машин квадратично зростає зі збільшенням швидкості. Закон зміни напруги живлення АД при зміні частоти $U = f^2 B^*$</p>
металообробні верстати		<p>Момент опору робочих машин зменшується при збільшенні швидкості. Закон зміни напруги живлення АД при зміні частоти $U = \sqrt{f} B^*$</p>

* B - постійний коефіцієнт.

Зміна частоти напруги живлення призводить до зміни швидкості АД і робочої машини. При цьому момент опору робочої машини також змінюється. Для того, щоб унеможливити перевантаження двигуна необхідно варіювати амплітуду його напруги. У табл. 1.3 приведені формули, за якими необхідно змінювати напругу живлення АД при зміні частоти його живлення для того, щоб перевантажувальна здатність двигуна була збережена [20].

1.2.5.2. Частотне векторне керування асинхронним двигуном

Сучасні системи векторного регулювання дозволяють просто й ефективно керувати такими складними об'єктами як асинхронний двигун із короткозамкненим ротором (АДКЗ). Це дозволяє істотно розширити область застосування асинхронного приводу і витіснити з автоматизованих керованих приводів двигуни постійного струму. Можливість використання векторних алгоритмів регулювання АД пов'язано в першу чергу з розвитком силової електроніки, яка дозволяє створювати надійні й відносно дешеві перетворювачі, а також з розвитком швидкодіючої мікроелектроніки.

Першим етапом процесу розвитку векторного керування АД була розробка універсальної векторно-матричної математичної моделі, що одержала назву узагальненої електричної машини [21]. Цей процес почався у кінці 20-х років і завершився наприкінці 40-х років ХХ століття. Модель узагальненої електричної машини дозволяє описувати електромагнітні процеси в ідеалізованому електродвигуні за допомогою апарату лінійної алгебри. Практичне використання моделі було відкладено на кілька десятиліть, оскільки при „ручних” розрахунках вона не давала переваг, але вимагала істотних обчислювальних робіт. Її успішно використовували для теоретичного аналізу перехідних процесів в електричних машинах.

Як відомо, повна керованість електропривода досягається тоді, коли забезпечується керування електромагнітним моментом двигуна. У електричних двигунах крутний момент утворюється в результаті взаємодії магнітних полів статора й ротора. Обидві величини повинні бути незалежні

одна від іншої. Тоді одну з них можна підтримувати постійною, а за допомогою іншої здійснювати регулювання. У двигунах постійного струму існують окремі електричні кола для керування магнітним потоком і моментом. Нагадаємо що електромагнітний момент машини постійного струму визначається за формулою (1.7). У АДКЗ є тільки один канал, у якому об'єднані обидві складові струму. Отже у завдання системи регулювання входить функція їхнього поділу. Це завдання вирішується при використанні рівнянь узагальненої електричної машини у векторній формі. У результаті вибору пари векторів величин, які утворюють електромагнітний момент, і системи координат, у якій вони представлені, можна одержати рівняння моменту у вигляді функції незалежних проєкцій цих величин на координатні осі [22-24]. Тоді керування моментом зводиться до керування проєкціями векторів. Звідси й назва способу.

В 1971 році Ф. Блашке (*F. Blaschke*) сформулював принцип керування, запатентований фірмою *Siemens* і названий трансвекторним керуванням (*TRANSVEKTOR®-Regelung*) [25]. Математичною основою його є рівняння електромагнітних процесів в АД у векторній формі. Рівняння справедливі для системи координат орієнтованої по напрямку магнітного поля. В англійській літературі цей принцип називається *field-oriented control (FOC)*, тобто „керування з орієнтацією по полю”. Він успішно використовується у наш час і повністю асоціюється з поняттям векторного керування, хоча в останній час із розвитком пристроїв обробки інформації з'явився інший спосіб, у якому також використовується векторне подання величин, але алгоритм керування відрізняється від трансвекторного. Цей спосіб називається прямим керуванням моментом (*DTC direct torque control*).

Для побудови систем векторного керування АД можуть бути використані будь-які пари векторів, за допомогою яких можна представити електромагнітний момент АД. Однак, від вибору векторів значною мірою залежить ступінь складності системи. Бажано, щоб величини, які представлені векторами в рівнянні моменту, можна було безпосередньо

виміряти й впливати на них при керуванні моментом. У АДКЗ є тільки дві такі величини – це напруга й струм статора, і тільки одна з них, а саме струм статора, може входити в рівняння моменту. Тоді іншою величиною може бути тільки струм ротора або яке-небудь потокозчеплення у двигуні. Струм ротора принципово неможливо виміряти, а пристрої його ідентифікації за іншими вимірюваними параметрами складні й ненадійні. Тому для вибору залишаються три потокозчеплення: статора, ротора й основне, тобто магнітний потік у зазорі АД. Потокозчеплення статора й робочий потік АД можна безпосередньо виміряти й використати цей сигнал у системі керування, що часто й робиться при створенні приводів. У масових виробках розробники намагаються використати сигнали, доступні без установки датчиків (струм і напруга статора). За миттєвим значенням цих величин можна обчислити, наприклад, потокозчеплення статора. Однак, при виборі потокозчеплення статора або основного потокозчеплення передаточні функції системи керування виходять досить складними й мало підходящими для практичного використання.

Простіший вигляд мають рівняння електромагнітних процесів в АД у випадку подання їх через вектор потокозчеплення ротора ψ_2 . У системі координат орієнованій по потокозчепленню ротора (ця система координат названа dq -системою, вона обертається у просторі зі швидкістю ротора) рівняння електромагнітного моменту буде мати вигляд:

$$M = C\psi_2 I_{1q}, \quad (1.9)$$

де C – деякий коефіцієнт; ψ_2 - потокозчеплення ротора; I_{1q} - проекція вектора струму статора на вісь q , яка перпендикулярна потокозчепленню ротора ψ_2 (рис. 1.23).

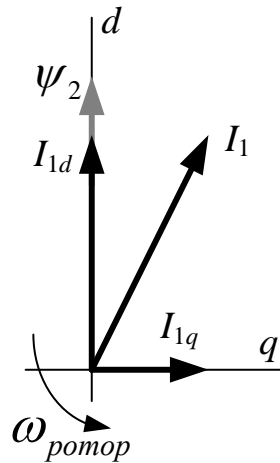


Рис. 1.23. Вектори, які визначають електромагнітний момент АД у системі координат dq

Рівняння (1.9) аналогічне рівнянню (1.7): ψ_2 виконує роль основного магнітного потоку, а I_{1q} - струму якоря двигуна постійного струму.

Таким чином, принцип векторного керування полягає у роздільному керуванні магнітним потоком і моментом АД за допомогою незалежних складових струму статора, які відповідають проєкціям вектора струму на осі системи координат dq , орієнтованої за напрямком вектора магнітного потоку ротора.

Отже, якщо задати значення поздовжньої складової струму, що відповідає необхідному магнітному потоку, а поперечної – необхідному моменту на валу, то тим самим буде визначено вектор струму статора в системі координат dq . Після цього здійснюється перетворення системи координат dq у нерухому і розкладання вектора струму на фазні проєкції, у результаті чого утворюються синусоїдальні сигнали, що відповідають фазним струмам, які потрібно сформувати в обмотках статора, щоб одержати заданий електромагнітний момент [26].

Перетворення систем координат (з нерухомої у рухому dq) неможливо без інформації про просторове положення опорного вектора ψ_2 у кожен момент часу. Цю інформацію можна одержати безпосереднім виміром

магнітного потоку статора або робочого потоку за допомогою датчиків, а потім обчислити ψ_2 . Однак при цьому необхідно включати у конструкцію приводу ненадійні датчики Хола. Уникнути небажаних (і не завжди можливих) змін конструкції АД можна використовуючи виміри інших параметрів:

- 1) струмів статора АД;
- 2) напруг на затискачах статора АД;
- 3) кутової швидкості ротора АД;
- 4) кутового положення ротора АД.

Для векторного керування АД необхідно вимірювати принаймні дві з перелічених величин.

Більш простим і дешевими є електропривод, розімкнутий по швидкості, який не містить датчика кутової швидкості (використовується інтернаціональний термін „безсенсорне керування”). Тут виміру підлягають лише струми й напруги статора АД. Швидкість ротора обчислюється в перетворювачі частоти (ПЧ) за математичною моделлю АД, що враховує усереднені типові або реально виміряні параметри АД. Модель АД представляє собою систему диференціальних рівнянь, які розв’язуються у реальному часі. Невідомими функціями часу в цій системі можуть бути вектори струмів або потокозчеплень ротора, статора і швидкість ротора. При вимірюванні параметрів і розв’язуванні диференціальних рівнянь у системі керування ПЧ накопичуються похибки, що приводить до неточного визначення орієнтації поля. Для стійкої роботи застосовується безперервна корекція розв’язків системи на основі різниць виміряних і обчислених значень струмів статора.

Через складність точного відображення параметрів АД (наприклад, внаслідок зміни температурних умов роботи АД) регульовальні властивості таких векторних безсенсорних електроприводів у зоні нульових швидкостей різко погіршуються, тобто має місце обмеження діапазону регулювання. Для роботи в необмеженому діапазоні швидкостей використовуються замкнуті по

швидкості електроприводи, які мають електромеханічний або цифровий датчик швидкості (енкодер), встановлений на валу АД. У цьому випадку модель АД значно спрощується й служить тільки для визначення орієнтації потокозчеплення ротора. Ця інформація використовується для перетворень координат.

Векторне керування АД, особливо без датчика швидкості, вимагає великого обсягу й високої швидкості обчислень, що може бути реалізовано потужним контролером, наприклад, цифровим сигнальним DSP-контролером [27].

Рациональні області застосування асинхронних електроприводів з векторним керуванням:

- спеціальні верстати, промислові роботи в тому числі із задачами позиціонування робочого органу;
- привод з частими пусками, зупинками, реверсами, наприклад, у сфері підйомно-транспортного обладнання (транспортні механізми, ліфти тощо);
- електричний транспорт.

Реалізація зміни частоти для керування швидкістю двигуна виконується за допомогою перетворювача частоти.

1.2.5.3. Будова та принцип дії частотного перетворювача

Силова частина ПЧ складається з вхідного випрямляча, ланки постійного струму, інвертора та системи керування [28] (рис. 1.24). Вхідний випрямляч виконаний за схемою Ларіонова. Випрямляч приєднаний до ланки постійного струму, яка складається з термісторів R_1 і R_2 та електролітичних конденсаторів великої ємності $C1$ і $C2$. Термістори, з від'ємним температурним коефіцієнтом, призначені для обмеження зарядного струму при вмиканні перетворювача. В холодному стані термістори мають великий опір. Після вмикання перетворювача в мережу струм, що протікає через термістори, розігріває їх, внаслідок чого вони різко зменшують свій опір.

Таким чином, термістори обмежують зарядний струм конденсаторів, а в процесі роботи практично не впливають на величину напруги конденсаторів. Конденсатори $C1$ і $C2$ згладжують напругу ланки постійного струму, фільтрують споживаний з мережі струм, забезпечують повернення реактивної енергії двигуна при вимиканні інвертора та при перехідних процесах і забезпечують повернення енергії при частотному гальмуванні двигуна M . Резистори $R2$ і $R3$, увімкнені паралельно кожному конденсатору, зрівнюють їх заряд. До виходу ланки постійного струму приєднано трифазний інвертор напруги, який складається з трьох півмостів на транзисторах $VT2 - VT7$, зворотних діодів $VD2 - VD7$ та гальмівного ключа на транзисторі $VT1$ зі зворотним діодом $VD1$. До гальмівного ключа при необхідності приєднують гальмівний резистор, моду буде розсіювати енергію в режимі частотного гальмування двигуна. Зворотні діоди $VD2 - VD7$ служать для передачі енергії від двигуна до джерела живлення (у ланку постійного струму). Основною особливістю інвертора представленого на рис. 1.24 є те, що напруга при навантаженні визначається лише черговістю комутації транзисторів інвертора. Досягається це введенням шунтуючих діодів $VD2 - VD7$ для проведення індуктивного струму навантаження. В результаті цього інвертор має жорстку зовнішню характеристику, тобто при зміні навантаження напруга інвертора практично не змінюється (такий інвертор називається інвертором напруги). Інвертори можуть комутуватись від електромережі змінного струму і автономно за допомогою спеціальних блоків, що мають конденсатори й котушки індуктивності. В частотно-регульованих приводах, як правило, використовуються автономні інвертори.

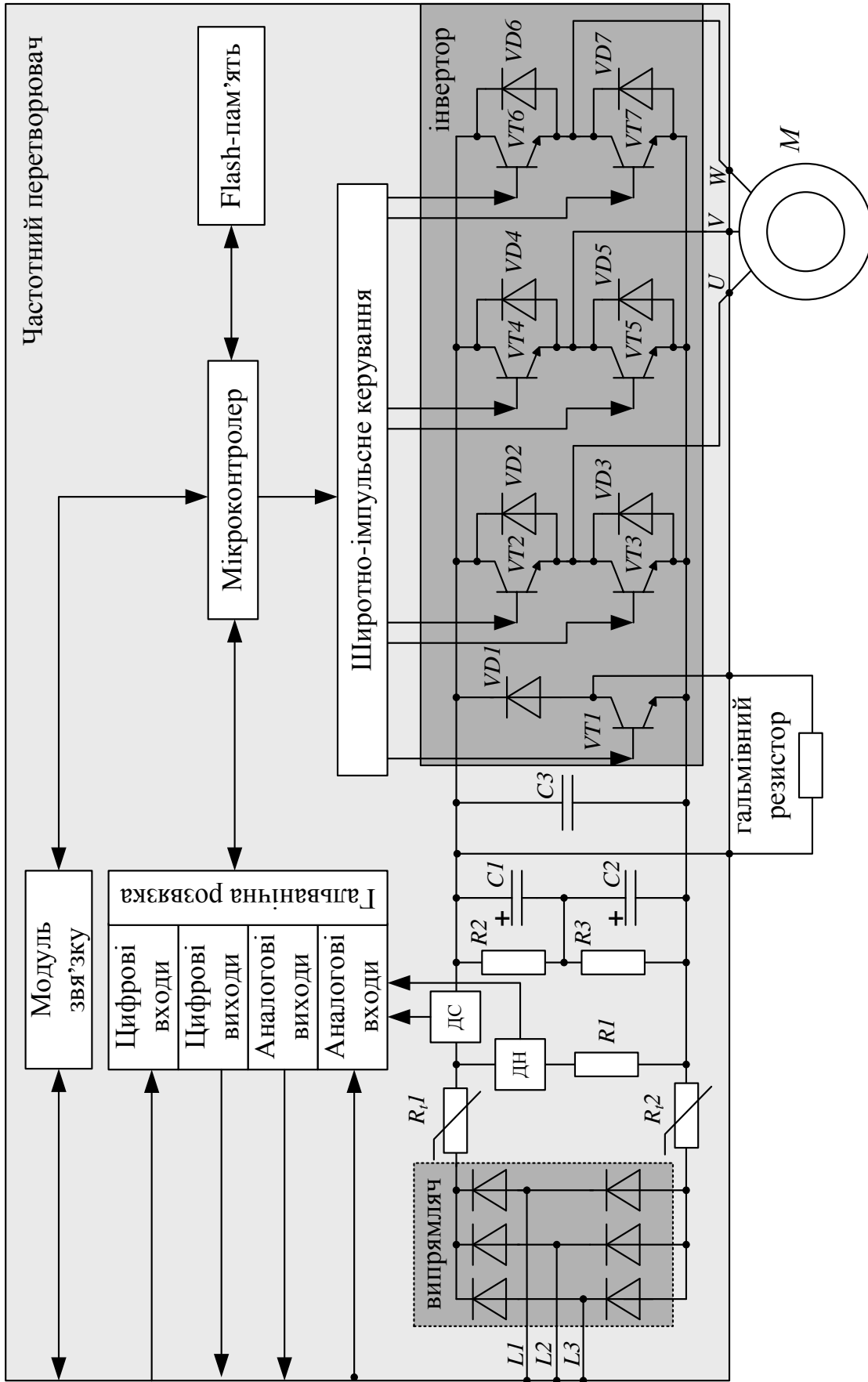


Рис. 1.24. Схема ПЧ

Через велику частоту комутацій ключів інвертора (близько 16 кГц) між додатною та від'ємною шиною ланки постійного струму виникають великі перенапруги з частотою вільних коливань в кілька мегагерц. Для захисту силового модулю від пробою до його затискачів приєднано так званий снабберний конденсатор малої ємності $C3$. Ємність цього конденсатора в десятки разів менша ємності електролітичних конденсаторів, що дозволяє ефективно боротися з перенапругами. Керування інвертором здійснюється від мікропроцесора шляхом широтно-імпульсної модуляції напруги. Транзистори інвертора працюють у ключовому режимі. Якщо коефіцієнт заповнення імпульсу в періоді змінювати за синусоїдним законом, то так же буде змінюватися і середнє значення напруги фази в періоді. На входи мікропроцесора подаються аналогові або цифрові сигнали від зовнішніх керуючих пристроїв, датчиків напруги (ДН) і струму (ДС) тощо.

Одними з найбільш поширених пристроїв сучасної силової електроніки є автономні інвертори напруги (АІН). У складі ПЧ вони перетворюють постійну напругу у змінну регульованої частоти і амплітуди. Вони також є основними блоками пристроїв безперебійного живлення. Оскільки вихідна напруга АІН складається з прямокутних імпульсів, то формування необхідної частоти і величини напруги здійснюється за рахунок модуляції вихідних імпульсів. В даний час найбільше поширення отримала широтно-імпульсна модуляція (ШІМ), що виражається у впливі на тривалість імпульсів і моменти їх появи.

Опис процесів, що відбуваються за період вихідної напруги при багаторазовій комутації швидкодіючих силових ключів, послідовність роботи таких ключів при формуванні напруги на виході ПЧ, прикладеної до статора АД буде дано у цьому пункті.

Розглянемо процеси, що відбуваються при роботі трифазного АІН, зібраного по мостовій схемі і підключеного до симетричного активно-індуктивного $R-L$ навантаження, яким моделюється АД (рис. 1.25).

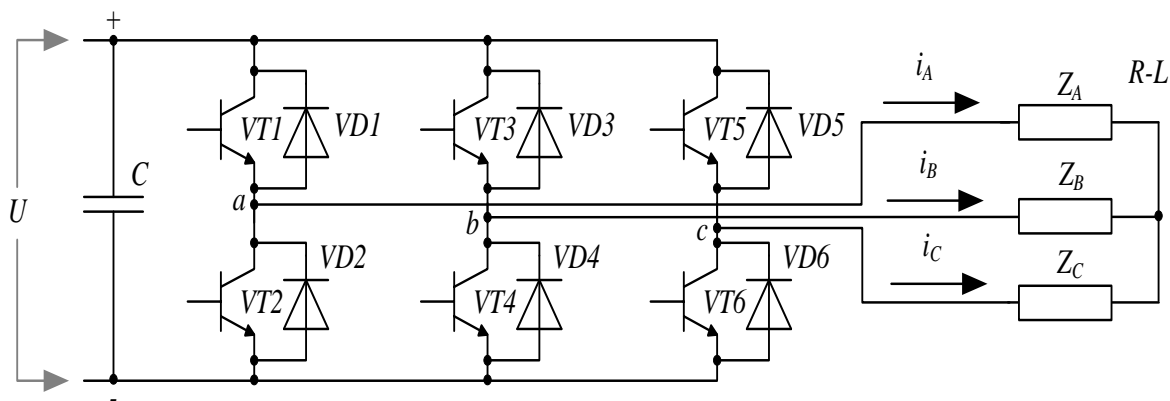


Рис.1.25. Схема трифазного АІН з активно-індуктивним навантаженням

У кожному плечі моста інвертора зустрічно-паралельно включені біполярний транзистор з ізольованим затвором IGBT ($VT1-VT6$) та швидкодіючі зворотні діоди ($VD1-VD6$). Додатні напрямки фазних струмів i_A , i_B та i_C в $R-L$ навантаженні вказані стрілками. До шини з додатною полярністю напруги U (U - напруга на конденсаторі C ланки постійного струму) приєднані колектори трьох транзисторів і катоди трьох зворотних діодів; ці вентиля зазвичай називають верхніми, на рис. 1.25 їм присвоєні непарні (1, 3 і 5) номери. До шини з від'ємною полярністю напруги під'єднані емітери (аноди) трьох нижніх, позначених парними (2, 4 і 6) номерами транзисторів (зворотних діодів).

Припустимо, що вентиля інвертора є ідеальними ключами, тобто володіють нульовим часом перемикання і, завдяки цьому, дозволяють здійснити багаторазову комутацію за період вихідної напруги інвертора, що формується за методом ШІМ. Крім того, для полегшення аналізу роботи АІН припускають, що ключі мають нульовий опір у замкненому стані і нескінченний опір – у розімкненому. Таким чином, схему представлену на рис. 1.25 можна подати у іншому (більш простому) вигляді (рис. 1.26).

На рис. 1.26 транзисторно-діодний модуль включений у одне плече моста інвертора позначено як звичайний ключ, який має двосторонню провідність. Усі ключі позначені цифрами: верхні – непарними, нижні – парними.

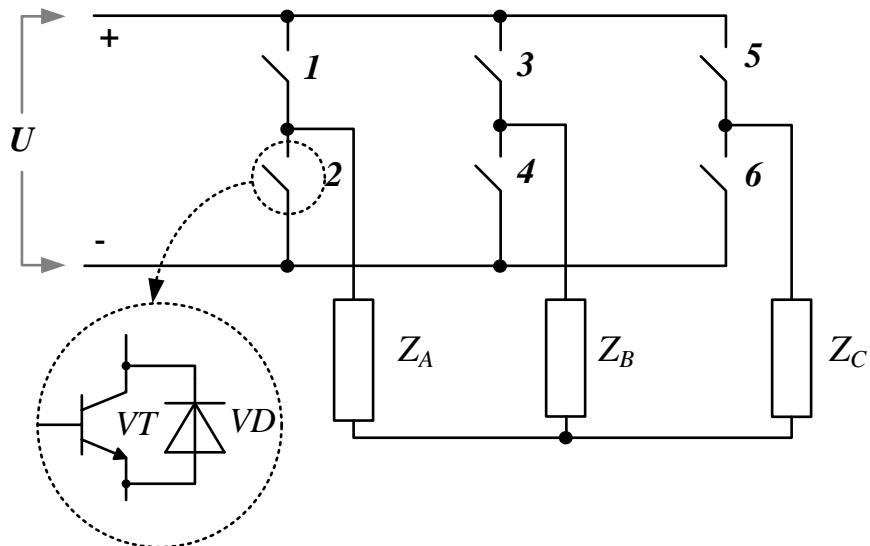


Рис.1.26. Спрощена схема трифазного АІН з активно-індуктивним навантаженням

У реальних схемах АІН пряма провідність кожного із силових керованих ключів (у напрямку від додатного до від'ємного полюса джерела живлення інвертора) забезпечується керованим напівпровідниковим приладом (тиристором або транзистором), а зворотна провідність - зворотним, як правило, некерованим вентиляем. Таким чином, якщо керований елемент силового ключа перебуває у включеному стані, то відповідне плече моста має двосторонню провідність струму. Якщо ж керований елемент (ключ) перебуває у виключеному стані, то відповідне плече моста має лише зворотну провідність (струм може протікати лише від АД до ланки постійної напруги).

Інвертування, тобто перетворення постійної напруги джерела живлення в трифазну змінну напругу необхідної частоти на вихідних затискачах А, В, С інвертора здійснюється комутацією ключів у плечах моста з певною частотою й у певній послідовності. Форма вихідної напруги інвертора визначається насамперед обраним законом перемикування ключів - законом комутації.

Розглянемо основні особливості робочих процесів у трифазному інверторі напруги при роботі його на АД у сталих режимах. На рис. 1.27

представлена діаграма станів трифазного мостового інвертора, яка показує порядок перемикання його ключів відповідно до закону комутації. Інтервали часу, протягом яких існує симетрична двостороння провідність (тобто провідність від АД до ланки постійної напруги та у зворотному напрямку, причому провідності ключа у обох напрямках однакові) відповідних плечей моста, виділені на діаграмі станів жирними горизонтальними лініями. Індекс горизонталі відповідає позначенню силового керованого ключа. Відповідно до показаної на діаграмі послідовності перемикання кожний із ключів перебуває у включеному стані протягом напівперіоду вихідної частоти (π). Під термінами „тривалість включеного стану” і „інтервал провідності ключа” мається на увазі саме інтервал часу, протягом якого існує симетрична двостороння провідність ключа й відповідного плеча моста. З діаграми рис. 1.27 видно, що протягом періоду робочої частоти ($0 \dots 2\pi$) інвертор перебуває в шести станах (I - VI), які послідовно змінюють один одного. Кожен з таких станів характеризується певною комбінацією включених і виключених ключів, отже для кожного стану характерна „своя” комбінація підключення навантаження до ланки постійної напруги АІН.

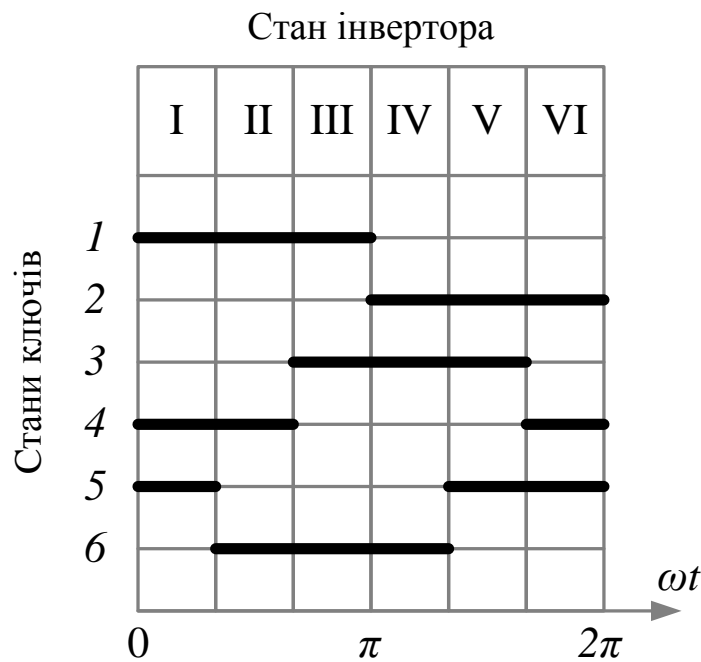


Рис. 1.27. Діаграма станів і графіки лінійних напруг трифазного АІН

У кожному зі станів інвертора будь-яка пара ліній живлення навантаження (наприклад, А і В) підключена або до різнополярних шин джерела живлення (до „+” та „-”), або до однієї із шин джерела й замкнута накоротко через ключі інвертора. У першому випадку лінійна напруга на виході ідеалізованого інвертора дорівнює напрузі джерела живлення U (це напруга у ланці постійного струму), а в другому - нулю.

Відповідні графіки лінійних напруг на виході інвертора показані на рис. 1.28 Якщо обмотки АД з'єднані за схемою „трикутник”, то лінійні напруги одночасно є фазними. Графіки лінійних напруг ідеалізованого інвертора при такому законі комутації ключів являють собою знакозмінні імпульсні функції, амплітуда яких дорівнює величині напруги джерела живлення, а частота дорівнює частоті комутації ключів. Як видно з рис. 1.28, графік лінійної напруги автономного інвертора є несинусоїдальним. Він може бути представлений у вигляді суми окремих складових (гармонік), причому такий розклад функції напруги у ряд Фур'є не буде містити гармонік кратних двом та трьом.

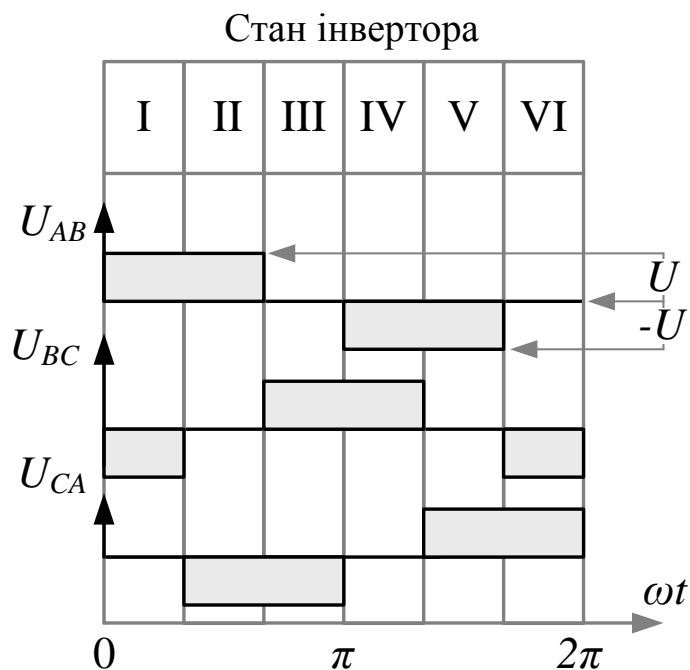


Рис. 1.28. Графіки лінійних (фазних) напруг на виході ПЧ при з'єднанні обмоток АД за схемою „трикутник”

Амплітуда лінійної напруги основної гармоніки буде трохи більша ніж напруга U (у ланці постійної напруги):

$$U_{л.осн.маx} = \frac{2\sqrt{3}}{\pi} U \approx 1,1U, \quad (1.10)$$

де $U_{л.осн.маx}$ - амплітуда основної гармоніки лінійної напруги на виході ПЧ; U - напруга ланки постійного струму ПЧ. Діюче значення лінійної напруги основної гармоніки:

$$U_{л.осн} = \frac{\sqrt{6}}{\pi} U \approx 0,78U. \quad (1.11)$$

Амплітуди вищих гармонік напруги зворотно пропорційні їх номерам, тобто чим вища гармоніка, тим меншу амплітуду вона має. Діюче значення вихідної лінійної напруги ПЧ:

$$U_{л} = \frac{\sqrt{2}}{\sqrt{3}} U \approx 0,81U. \quad (1.12)$$

Доля основної гармоніки у діючому значенні вихідної напруги ПЧ:

$$\frac{U_{л}}{U_{л.осн.}} = \frac{3}{\pi} \approx 0,955. \quad (1.13)$$

У випадку з'єднання обмотки АД в „зірку” графіки фазних напруг можуть бути знайдені з аналізу схем живлення фаз навантаження, наведених для кожного зі станів інвертора на рис. 1.29. Тут показані тільки ті ключі, які перебувають у провідному стані. Напруга ланки постійного струму розподіляється між фазами навантаження (у випадку їхньої симетрії) у такий спосіб: одна третина величини напруги спадає на паралельному з'єднанні фаз навантаження (АД) і дві третини - на послідовно включену фазу. Пояснимо такий розподіл напруг на фазах АД. Електрично АД представляє собою трифазне симетричне навантаження – його обмотки мають однакові повні

опори ($Z_A = Z_B = Z_C = Z$). Еквівалентний опір з'єднання обмоток, наприклад, для стану інвертора I такий:

$$Z_{екв} = \frac{Z_A Z_B}{Z_A + Z_B} + Z_C = \frac{ZZ}{Z + Z} + Z = \frac{Z^2}{2Z} + Z = \frac{Z}{2} + Z = \frac{3Z}{2}. \quad (1.14)$$

Струм, який протікає по такому еквівалентному опорі визначається так:

$$i = \frac{2U}{3Z}. \quad (1.15)$$

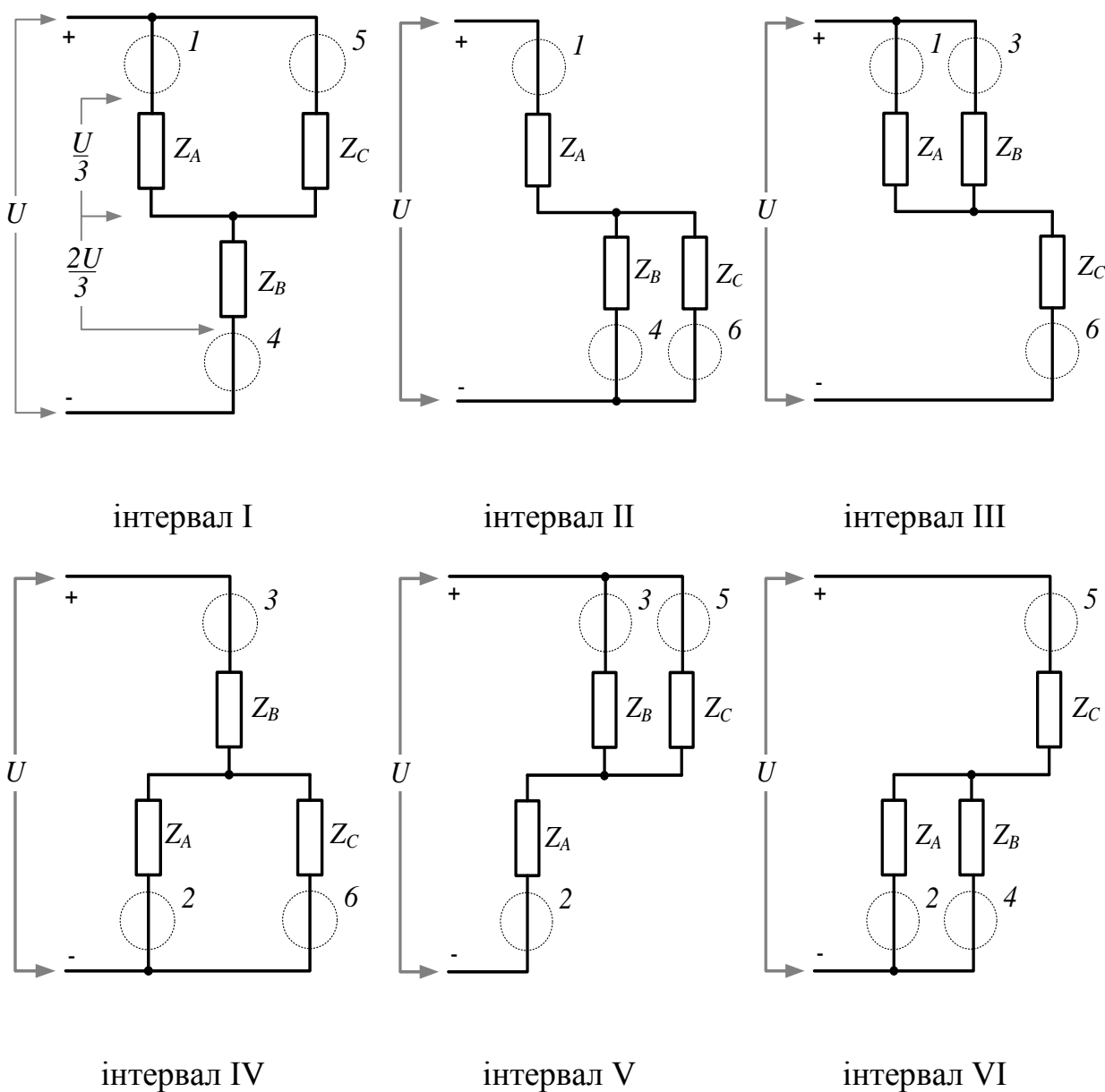


Рис. 1.29. Схеми живлення обмоток АД при їх з'єднанні за схемою „зірка”

Тепер визначимо спад напруги на опорі Z_B , який включений послідовно:

$$U_B = iZ_B = \frac{2U}{3Z} Z = \frac{2U}{3}. \quad (1.16)$$

Отже дві третини напруги спадає на послідовно включеній фазі АД (на фазі В), решта напруги (одна третина) – на паралельному з'єднанні фаз А і С двигуна, тобто до кожної паралельно включеної фази прикладається одна третина напруги ланки постійного струму ПЧ.

На рис. 1.30 представлені графіки фазних напруг на виході ПЧ, які відповідають прийнятому закону комутації ключів 1-6.

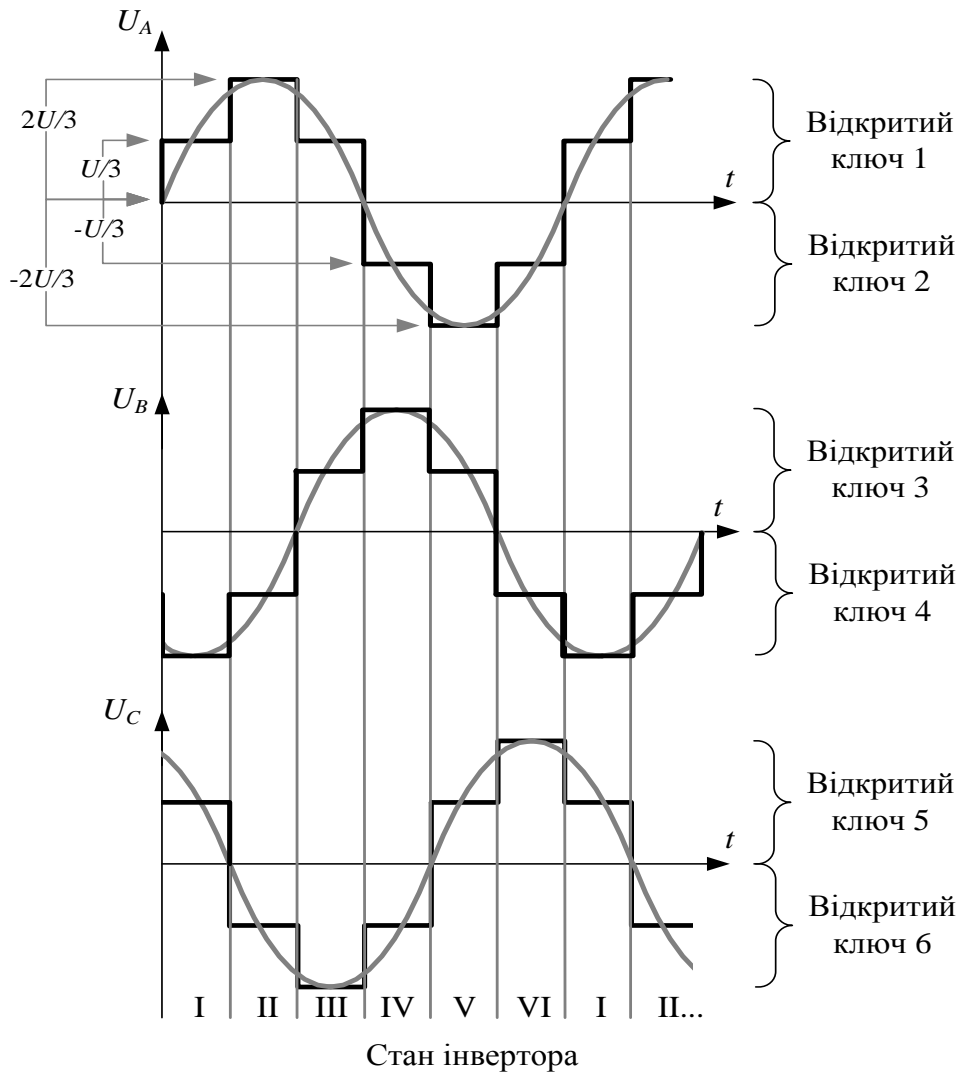


Рис. 1.30. Графіки фазних напруг на виході ПЧ при з'єднанні обмоток АД за схемою „зірка”

Несинусоїдальні графіки фазних напруг можуть бути представлені у вигляді суми окремих гармонічних складових за виключенням гармонік кратних двом та трьом. На рис. 1.30 лінією сірого кольору показана основна гармоніка фазної напруги. Визначити амплітудне та діюче значення основної гармоніки можна використавши відомі формули (1.10), (1.11). Для цього їх необхідно поділити на $\sqrt{3}$. Амплітудні та діючі значення вищих гармонік також необхідно поділити на $\sqrt{3}$.

Отже, доля основної гармоніки у діючому значенні напруги на виході ПЧ при перемиканні обмоток з „трикутника” на „зірку” не зміниться, тобто вираз (1.13) справедливий для будь-якої схеми з'єднання обмоток АД [21].

1.2.5.4. Основні функції частотного перетворювача

Питання про необхідність тих або інших функцій ПЧ сьогодні вкрай актуальне оскільки щороку на ринку з'являються нові вироби з більшими можливостями. Складно зрозуміти, який набір функцій зможе задовольнити потреби того або іншого технологічного процесу. Розглянемо основні функції ПЧ, пропоновані виробниками даної техніки. Існують певні відмінності між вбудованими й додатковими функціями ПЧ. Відомо, що частина функцій, як правило, входить у стандартне виконання ПЧ. Вартість цих функцій закладена в ціну виробу. У даному пункті розглянемо базові функції ПЧ. Зазначимо, що додаткові опції і пристрої ПЧ, наявність яких визначається з врахування конкретних технологічних процесів, дуже широкі. До базових функцій ПЧ відноситься [29]:

- реалізація різних методів керування електродвигуном (векторне та скалярне). Скалярне керування найпоширеніше й максимально задовольняє вимогам таких механізмів, як насоси, вентилятори, компресори, а також таких, для яких важливо підтримувати швидкість обертання або який-небудь технологічний параметр. Метод досить простий, але має невеликий діапазон регулювання швидкості. Метод DTC (із прямим керуванням моментом, без установки додаткових

датчиків) ефективно використовується при відносно невисоких вимогах до точності підтримки швидкості з діапазоном регулювання швидкості $D=700:1$, тобто для таких механізмів, як поршневі компресори, насоси, піднімальні механізми, конвеєри, дробарки, пилки, міксери тощо;

- ПД-регулятор. Вбудований регулятор використовується для керування зовнішнім процесом за допомогою сигналу зворотного зв'язку. Сигнал задання може надходити через аналоговий вхід, з панелі керування за допомогою задання або через послідовний інтерфейс. При цьому вимірюється відхилення величини, яка потребує стабілізації (наприклад, тиск, швидкість, температура тощо), від заданого значення і генерується керуючий сигнал. Наявність даного регулятора всередині ПЧ дозволяє спростити систему керування й відмовитися від використання зовнішніх регуляторів (контролерів). Наявність ПД-регулятора необхідно для таких механізмів, як насоси, верстати, транспортери тощо, тобто там, де потрібна точна підтримка контрольованої величини;
- аналогові та цифрові входи і виходи. Аналогові виходи дозволяють більш наочно представити значення якого-небудь параметра. Наприклад, їх часто використовують для відображення на стрілочному вольтметрі частоти обертання двигуна. Аналогові входи дозволяють подавати сигнали від датчиків прямо в ПЧ, без використання яких-небудь додаткових пристроїв. Як правило, використовуються уніфіковані типи сигналів (0-5 В, 0-10 В, 4-20 мА). Дискретні входи дозволяють керувати ПЧ із кнопок, установлених на лицьовій панелі шафи або з поста керування. Як правило, ці входи використовуються для подачі команд типу „ПУСК”, „СТОП”, „РЕВЕРС” або для ступінчастого перемикання швидкостей двигуна (рис. 1.31).

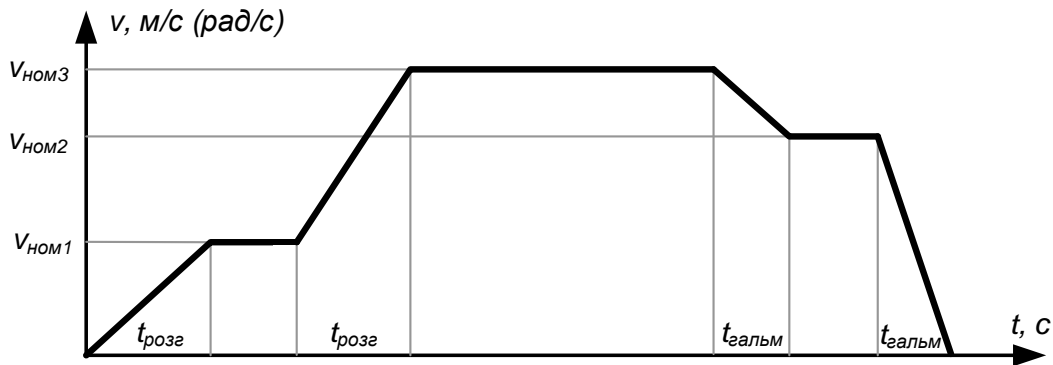


Рис. 1.31. Керування швидкістю двигуна за допомогою дискретних входів

Натискаючи кнопки „ПУСК”, „НОМІНАЛЬНА ШВИДКІСТЬ 1”, „НОМІНАЛЬНА ШВИДКІСТЬ 2”, „НОМІНАЛЬНА ШВИДКІСТЬ 3”, „СТОП” оператор технологічного процесу може змінювати швидкість обертання АД та зупиняти двигун дистанційно. До дискретних виходів також можна підключати індикаторні лампочки „Аварія”, „Обертання вперед”, „Обертання назад” тощо;

- інтерфейси зв’язку. У більшості сучасних ПЧ передбачена можливість підключення в промислову мережу. Звичайно використовуються протоколи RS-485 (RS-232), Modbus, Profibus, Interbus, CANOpen, DeviceNet. З’єднання перетворювачів у мережу дозволяє побудувати більш складну систему автоматичного керування технологічними процесами з використанням контролерів і промислових комп’ютерів;
- моніторинг навантаження (захист двигуна від механічного перевантаження/недовантаження) дозволяє використовувати ПЧ як пристрій для контролю навантаження і захисту двигуна від механічних перевантажень і недовантажень, наприклад, від заклинювання полотна конвеєра, шнекового транспортера, обриву паса вентилятора, „сухої” роботи насоса тощо. З появою ненормального навантаження двигуна ПЧ може зупинити двигун і здійснити затримку перед повторним включенням або подати аварійний сигнал. Використання цієї функції дозволяє уникнути дорогої установки додаткових датчиків (оскільки

для останніх потрібна установка безпосередньо в технологічний процес). Ця функція особливо необхідна для насосів, кранів, змішувачів, гвинтових конвеєрів, стрічкових конвеєрів, міксерів, дробарок тощо;

- взаємодія із системою автоматизації верхнього рівня. Здійснюється за принципом „ведучий - ведений” (Master - Slave), причому ПЧ виступає в ролі веденого пристрою. Програмне забезпечення реалізує необхідний протокол обміну, забезпечує прийом і виконання команд керування, а також видачу необхідної інформації про поточний режим роботи, стан датчиків і параметрів ПЧ. У випадку двопроцесорної системи забезпечується можливість спільної роботи основного контролера ПЧ і контролера пульта керування;
- діагностика апаратури і самодіагностика. Діагностика полягає у визначенні працездатності різних модулів, що входять до складу ПЧ (як силових, так і керуючих) і підключеного електродвигуна. Крім того, проводиться контроль цілісності програми й даних, які збуригаються в енергонезалежній пам'яті;
- реалізація захисних функцій. Сучасні ПЧ реалізують максимальний струмовий захист, захист від перегріву двигуна і перетворювача, від перевантаження, надмірних відхилень напруги живлення, обриву фази, міжфазного короткого замикання, замикання фази на землю й помилок зв'язку. Коректний вихід зі стану аварії можливий, тільки якщо ліквідовані причини її виникнення. Для окремих видів аварій (наприклад, провалу напруги в мережі, аварії зв'язку) система в стані самостійно відстежити можливість продовження роботи. Відновлення працездатності системи після інших аварій вимагає втручання обслуговуючого персоналу;
- збереження інформації про режими та тривалість роботи, періодичність включення ПЧ, ведення журналу збоїв і аварій, що відбулися. Це

дозволяє проаналізувати ефективність використання ПЧ й полегшує пошук причин збоїв у роботі обладнання;

- плавний пуск і зупинка двигуна з вибором форми кривої зміни швидкості (звичайно використовують лінійну, S- і U-подібну характеристики зміни швидкості) і роздільним налаштуванням часу розгону і гальмування (у межах від 0,1 с до 9999 с з дискретністю 0,1 с) з автоматичною корекцією прискорення у випадку перевищення допустимого моменту (ця опція дозволяє задавати мінімальну тривалість розгону та гальмування без необхідності перевірки умови перевищення максимального струму ПЧ. На рис. 1.32 показані різні варіанти графіків функції зміни швидкості двигуна.

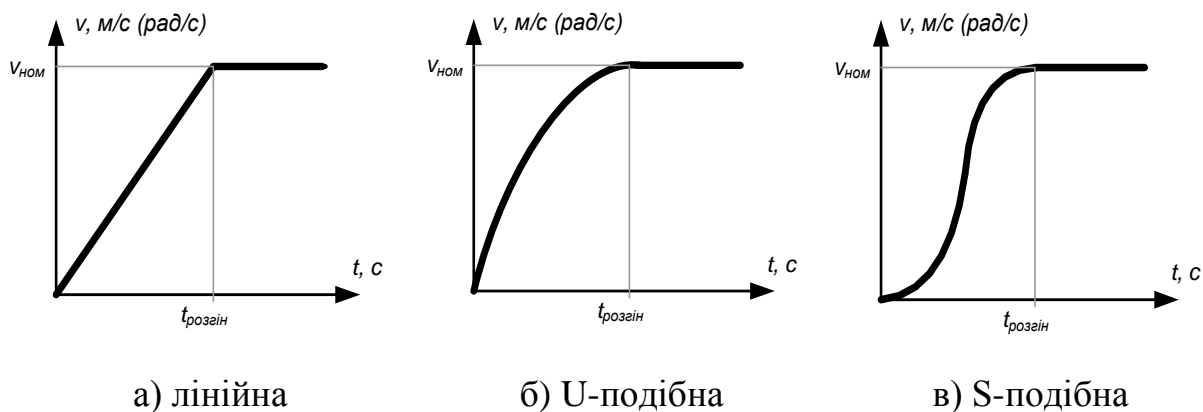


Рис. 1.32. Діаграми зміни швидкості двигуна при різних характеристиках розгону

У випадку якщо таке перевищення має місце ПЧ самостійно збільшує тривалість розгону/гальмування АД);

- режим „підхоплення” електродвигуна, використовуваний при включенні ПЧ на обертовий двигун (наприклад, після короткочасного провалу напруги мережі живлення). Ця функція дозволяє зменшити стрибки струму і напруги при пуску двигуна у момент „підхоплення”. Зменшується також початковий (пусковий) момент АД. Реалізація цієї функції полягає у тому, що у момент запуску ПЧ „підбирає” частоту з якою обертається АД;

- режим гальмування (на вибігу, частотне гальмування, гальмування постійним струмом). Найпростіший тип гальмування двигуна – на вибігу – відбувається від дії моментів опору у приводі та робочій машині. При цьому ПЧ не керує двигуном. Частотне гальмування АД полягає у поступовому зменшенні частоти напруги живлення АД (рекуперативне гальмування). Гальмування постійним струмом (динамічне гальмування) часто використовується для „догальмовування” двигуна на низькій швидкості;
- пропуск частот, при яких робота електропривода небажана, наприклад, через ризик виникнення механічних резонансів. Це особливо важливо для приводів відцентрових насосів і вентиляторів, в яких явище резонансу виникає особливо часто. Графік зміни частоти ПЧ представлений на рис. 1.33 ілюструє пропускання „вікон” при розгоні двигуна. Таких „вікон” може бути декілька, їх ширина, як правило, становить 5 Гц;

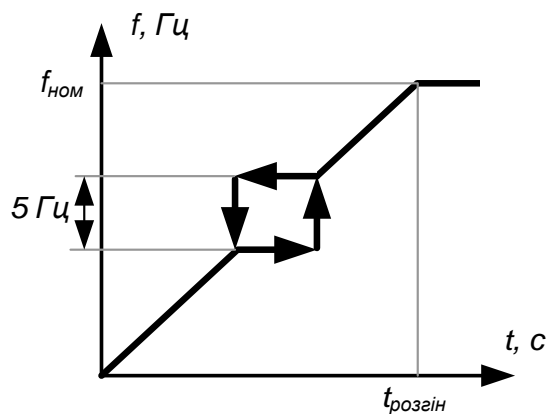


Рис. 1.33. Зміна частоти на виході ПЧ з пропуском „вікна” шириною 5 Гц

- фіксовані набори параметрів. Велика кількість параметрів дає користувачеві можливість більш гнучко налаштувати ПЧ під необхідні завдання. Цю функцію зручно використовувати, коли потрібно поміняти режим роботи двигуна. Це здійснюється шляхом вибору в меню набору параметрів, які відповідають потрібному режиму роботи.

Тобто один ПЧ може працювати з декількома електродвигунами різної потужності, що функціонують з різними робочими машинами;

- підтримка високого пускового моменту на низьких частотах за рахунок додаткового збільшення напруги;
- автоматичне визначення параметрів підключеного електродвигуна (індуктивність, опір обмоток тощо);
- наявність інтуїтивно-зрозумілого меню, мова меню. Багато виробників найчастіше пропонують уніфіковане англomовне меню. Така додаткова доробка, як введення інших мов у меню (української, російської), дає користувачам можливість швидше розібратися з усіма налаштуваннями й параметрами, а також краще сприймати показання поточних параметрів на дисплеї.

Окрім основних (базових) функцій ПЧ мають інші функціональні можливості: керування декількома електродвигунами; здійснення роботи ПЧ за розкладом тощо. Насамкінець зазначимо, що більшість функцій ПЧ „скриті” від користувача, оскільки це функції „тонкого” налаштування ПЧ. Для їх активізації необхідно задіяти певну функцію ПЧ, яка дозволяє їх змінювати.

1.2.6. Модулі руху в мехатронних системах

Створення мехатронних модулів руху з високими характеристиками є однією з найважливіших завдань автоматизованого машинобудування. З таких модулів, як з функціональних блоків, можна створити складні мехатронні системи.

Введемо основні визначення й поняття [30].

Модуль руху (МР) – конструктивно й функціонально самостійний виріб, що включає в себе механічну (гідравлічну, пневматичну) і електротехнічну частини, який можна використовувати індивідуально й у різних комбінаціях з іншими модулями. Головною відмінністю МР від загальнопромислового приводу є використання вала двигуна в якості одного

з елементів механічного перетворювача руху. Прикладами МР є мотор-редуктор, мотор-колесо, мотор-барабан, електрошпіндель.

Мехатронний модуль руху (ММР) – конструктивно й функціонально самостійний виріб, що включає в себе механічну (гідравлічну, пневматичну), електротехнічну, електронну й інформаційну частини, який можна використовувати індивідуально й у різних комбінаціях з іншими модулями. На відміну від модулів руху (МР) у мехатронних модулях руху (ММД) присутнє також електронне й інформаційне обладнання.

Інтелектуальний мехатронний модуль (ІММ) – конструктивно й функціонально самостійний виріб із синергетичною інтеграцією механічної (гідравлічної, пневматичної), електротехнічної й комп'ютерної (мікропроцесорної) частин, який можна використовувати індивідуально й у різних комбінаціях з іншими модулями. Таким чином, у порівнянні із ММР, у конструкцію ІММ додатково включені мікропроцесорне обчислювальне обладнання й силові електронні перетворювачі, а також елементи, що забезпечують інтелектуальне керування.

1.2.6.1. Електромеханічні модулі руху

В 1927 р. фірмою „Бауер” була розроблена принципово нова конструкція – мотор-редуктор, що об'єднала в один компактний конструкційний модуль електродвигун і механічний перетворювач руху (редуктор). З тих пір з'явилася величезна гама різних мотор-редукторів для різних умов застосування, які дозволяють знайти оптимальне вирішення завдання у кожному конкретному випадку. Конструктивне об'єднання електродвигуна й перетворювача руху в єдиний компактний електропривод – мотор-редуктор – має ряд переваг у порівнянні із застарілою системою з'єднання електродвигуна й перетворювача руху через муфту: значне скорочення габаритних розмірів, істотне зменшення кількості приєднувальних деталей, зменшення витрат на установку, налагодження й

запуск. Таким чином, мотор-редуктор є в цей час одним з найпоширеніших видів електропривода.

1.2.6.2. Мехатронні модулі руху

Вбудовані в модулі руху (МР) мініатюрні датчики й електронні блоки для обробки їх сигналів перетворюють МР у мехатронні модулі руху (ММР) [30]. Для створення сучасних технологічних машин, призначених для автоматизованого машинобудування, необхідні різноманітні мехатронні модулі руху, що задовольняють ряду вимог: висока точність реалізації рухів, надійність, довговічність, можливість роботи при наявності різних видів збурень і в широкому діапазоні температур навколишнього середовища, а також значно менші масогабаритні показники у порівнянні звичайним електроприводом. Вимоги до приводних зусиль, точності й швидкості виконавчих рухів пов'язані з особливостями технологічної операції, яка обладнується мехатронними модулями руху, а вимога мінімізації його розмірів – необхідністю вбудовування його в технологічну машину. Спроба синтезу мехатронного модуля руху з наявних компонентів, що серійно випускаються, може привести до технічно й економічно неефективного вирішення завдання. Тому більш раціональним є проектування спеціалізованого мехатронного модуля руху, що найбільш повно відповідає його службовому призначенню. Складність і суперечливість вимог, поставлених до модулів руху, обумовлює доцільність мехатронного підходу до їхнього проектування. Зокрема, принцип синергетичної інтеграції елементів системи приводить до забезпечення бажаного рівня якості модуля за рахунок конструктивного й функціонального взаємопроникнення його компонентів, багато з яких є спеціалізованими й створюються в ході паралельного системного проектування з врахуванням їх наступного ефективного об'єднання. Якщо переважною є вимога забезпечення компактності мехатронного модуля руху, то вона може бути реалізована шляхом використання безконтактних електричних машин і їх інтеграції з

перетворювачами руху й інформаційно-вимірювальними елементами. При цьому перетворювачі руху й датчики не є окремим обладнанням, а стають невід’ємними елементами двигуна. Синергетичний ефект досягається також за рахунок виконання деякими компонентами мехатронного модуля руху декількох функцій одночасно. Таке вирішення завдання дозволяє виключити багато механічних інтерфейсів, спростити й здешевити конструкцію, усунути необхідність механічного „узгодження” датчика й двигуна. Мехатронні модулі руху є функціональними „цеглинками”, з яких можна компонувати складні мехатронні системи. Приклади мехатронних модулів руху: мехатронні модулі руху на основі електродвигунів кутового й лінійного руху й різних перетворювачів руху (гвинтових, черв’ячних, планетарних, хвильових тощо), безредукторні мехатронні модулі руху, безредукторні поворотні столи.

1.2.6.3. Інтелектуальні модулі руху

Як відзначалося вище, головною особливістю сучасного етапу розвитку мехатроніки є створення принципово нового покоління модулів – інтелектуальних мехатронних модулів (ІММ) [30]. Розглянемо основні переваги, які дає застосування інтелектуальних мехатронних модулів:

- ▶ здатність ІММ виконувати складні рухи самостійно, без „звертання” до верхнього рівня керування, що підвищує автономність модулів, гнучкість і живучість мехатронних систем, які працюють у змінних і невизначених умовах зовнішнього середовища;
- ▶ спрощення комунікацій між модулями й центральною системою керування (наприклад, перехід до безпроводних комунікацій), що дозволяє досягти підвищеної перешкодозахищеності мехатронної системи і її здатності до швидкої реконфігурації;
- ▶ підвищення надійності й безпеки мехатронних систем завдяки комп’ютерній діагностиці несправностей і автоматичному захисту в аварійних і позаштатних режимах роботи;

- ▶ створення на основі ІММ розподілених систем керування із застосуванням мережевих методів, апаратно-програмних платформ на базі персональних комп'ютерів і відповідного програмного забезпечення;
- ▶ використання сучасних методів теорії керування (програмних, адаптивних, інтелектуальних, оптимальних) безпосередньо на виконавчому рівні, що суттєво підвищує якість процесів керування;
- ▶ інтелектуалізація силових перетворювачів, що входять до складу ІММ, для реалізації безпосередньо в мехатронному модулі інтелектуальних функцій керування рухом, захисту модуля в аварійних режимах і діагностики несправностей;
- ▶ інтелектуалізація сенсорів для мехатронних модулів дозволяє досягти більш високої точності вимірів, програмним шляхом забезпечивши в самому сенсорному модулі фільтрацію шумів, калібрування, лінеаризацію характеристик „вхід-вихід”, компенсацію гістерезису й дрейфу нуля.

Основним фактором, що стримує використання інтелектуальних мехатронних модулів у серійних виробках, є їхня висока ціна, хоча в останні роки вона постійно знижується. Це обумовлене рядом технологічних факторів:

- ▶ бурхливим розвитком інформаційних технологій, орієнтованих на завдання керування рухом;
- ▶ появою напівпровідникових приладів нового покоління (силових польових транзисторів, біполярних транзисторів з ізольованим затвором);
- ▶ переходом на нову елементну базу в системах керування рухом (цифрові сигнальні DSP-процесори і блоки FPGA: Field Programmable Gate Arrays);

- розробкою гібридних технологій мехатроніки, що дозволяють вбудовувати електронне й обчислювальне обладнання в механічні вузли.

Інтелектуальний мехатронний модуль складається з наступних основних елементів:

- 1) електродвигуна (хоча можливе використання приводів і інших типів, наприклад, гідравлічних);
- 2) механічного перетворювача;
- 3) датчиків зворотного зв'язку й сенсорного обладнання;
- 4) керуючого контролера;
- 5) силового перетворювача;
- 6) обладнання зв'язку.

У сучасних ІММ використовують різні типи електродвигунів: кутового й лінійного руху, змінного й постійного струму, колекторні й вентильні, безперервного руху й крокові.

У якості механічних перетворювачів руху застосовують зубчасті, гвинтові та інші передачі. У конструкціях деяких ІММ, побудованих на базі високомоментних двигунів, перетворювачі руху відсутні. В інтелектуальних мехатронних модулях використовують різні датчики положення й швидкості (енкодери, обертові трансформатори, тахогенератори), датчики струму, моменту, температури, вібрації тощо, які передають інформацію в обладнання комп'ютерного керування про фактичний стан підсистем модуля.

Вбудовані керуючі контролери, реалізовані на сучасній елементній базі, дозволяють одержати компактні й надійні мехатронні вироби, що володіють інтелектуальними функціями, і будувати на їхній основі багатокоординатні мехатронні системи з децентралізованим керуванням. Це дозволяє створити якісно нові системи керування, що володіють принципово кращими характеристиками по швидкодії, точності й функціональній гнучкості, аніж вже відомі. Принципово важливо, щоб усі перераховані елементи були конструктивно об'єднані розроблювачем у єдиному корпусі.

При цьому обладнання зв'язку стають внутрішніми блоками, недоступними для користувача. В цілому можна виділити три напрямки інтелектуалізації мехатронних модулів руху:

- ▶ розвиток інтегрованих інтерфейсів, що зв'язують керуючий контролер з комп'ютером верхнього рівня в єдиний апаратно-програмний керуючий комплекс;
- ▶ створення інтелектуальних силових модулів керування шляхом інтеграції керуючих контролерів і силових перетворювачів;
- ▶ розробка інтелектуальних сенсорів мехатронних модулів, які додатково до звичайних вимірювальних функцій здійснюють комп'ютерну обробку й перетворення сигналів по гнучких програмах.

1.3. Контролерна та комп'ютерна техніка у мехатронних системах

1.3.1. Історичний розвиток мікроконтролерів та сфери їх застосування у сучасному виробництві

Мікроконтролер (англ. Micro Controller Unit, MCU) - мікросхема, що виконує керування електронними пристроями згідно записаних у нього алгоритмів з використанням інформації, одержуваної від датчиків. По суті, це однокристальний комп'ютер, здатний виконувати прості завдання. Слово „контролер” походить від англійського control (керувати), а не від слова „контроль” (облік, перевірка) [31].

Перші контролери з'явилися на рубежі 60-х і 70-х років в автомобільній промисловості, де використовувалися для автоматизації складальних ліній. У той час комп'ютери коштували надзвичайно дорого, тому контролери будувалися на твердій логіці (програмувалися апаратно), що було набагато дешевше. Однак переналаштування з однієї технологічної лінії на іншу вимагало фактично виготовлення нового контролера. Тому з'явилися контролери, алгоритм роботи яких міг бути змінений трохи простіше - за допомогою схеми з'єднань реле. Такі контролери одержали

назву програмованих логічних контролерів (ПЛК), і цей термін зберігся дотепер. Скрізь нижче терміни „контролер” і „ПЛК” ми будемо вживати як синоніми.

Перший патент на однокристальний мікроконтролер був виданий в 1971 році інженерам М. Кочрену й Г. Буну, співробітникам американської компанії Texas Instruments. Саме вони запропонували на одному кристалі розмістити не тільки процесор, але й пам'ять із пристроями вводу-виводу.

З появою потужних і дешевих мікроконтролерів в 1972 р. ринок ПЛК почав рости експоненціально й за період з 1978 по 1990 р.р. збільшився з 80 млн дол. до 1 млрд дол., а у 2002 р. склав 1,4 млрд дол. У цей час світовий ринок ПЛК продовжує рости, хоча й набагато меншими темпами.

В 1976 році американська фірма Intel випускає мікроконтролер i8048. Трохи пізніше в цьому ж році Intel випускає наступний мікроконтролер: i8051. Вдалий набір периферійних пристроїв, можливість гнучкого вибору зовнішньої або внутрішньої програмної пам'яті й прийнятна ціна забезпечили цьому мікроконтролеру успіх на ринку. З погляду технології мікроконтролер i8051 був для свого часу дуже складним виробом - у кристалі було використано 128 тис. транзисторів, що в 4 рази перевищувало кількість транзисторів в 16-розрядному мікропроцесорі i8086.

На сьогоднішній день існують сотні мікроконтролерів різноманітних фірм. Популярністю в розробників користуються 8-бітні мікроконтролери PIC фірми Microchip Technology й AVR фірми Atmel, 16-бітні MSP430 фірми TI, а також ARM, архітектуру яких розробляє фірма ARM і продає ліцензії іншим фірмам для їхнього виробництва.

Разом із конструктивним розвитком мікроконтролерів необхідно також висвітлити питання розвитку способів їх програмування. Як було зазначено вище, один із найперших способів програмування мікроконтролера – заміна схеми з'єднань реле. Таким способом програмувались мікроконтролери на перших етапах їх розвитку.

Пізніше з'явилися ПЛК, які можна було програмувати машинно-орієнтованою мовою, що було простіше конструктивно, але вимагало участі спеціально навченого програміста для внесення навіть незначних змін в алгоритм керування, що було незручно та дорого. Тому почалася боротьба за спрощення процесу програмування ПЛК, яка привела спочатку до створення мов високого рівня, а потім - спеціалізованих мов візуального програмування, схожих на мову релейної логіки. Цей процес завершився створенням міжнародного стандарту IEC 61131-3, який пізніше був перейменований у MEK 61131-3. Стандарт MEK 61131-3 підтримує п'ять мов технологічного програмування, що виключає необхідність залучення професійних програмістів при побудові систем з контролерами.

ПЛК використовуються практично у всіх сферах людської діяльності: для автоматизації технологічних процесів, у системах протиаварійного захисту й сигналізації, у верстатах із ЧПК, для керування дорожнім рухом, у системах життєзабезпечення будинків, для збору й архівування даних, у системах охорони, у медичному обладнанні, для керування роботами, у системах зв'язку, при постановці фізичного експерименту, для керування космічними кораблями, для автоматизації випробувань продукції тощо. Однак, дотепер залишається багато галузей економіки, куди контролерна техніка тільки починає проникати.

Згідно з останнім опитуванням в Інтернеті, проведеному журналом Control Engineering разом з Read Research, найбільше ПЛК використовується для керування верстатами. На другому місці - завдання, пов'язані з керуванням технологічними процесами, далі йде керування переміщеннями й завдання діагностики. Найчастіше контролери використовують для вирішення власних завдань (54 %), рідше для виробництва обладнання для продажу (25 %) або для того й іншого (17 %) [31].

Контролери використовуються не тільки як автономні засоби локального керування мехатронними установками, але й у складі

широкомасштабних систем автоматизованого керування цілими підприємствами (мехатронними комплексами).

Широкому поширенню ПЛК сприяє ріст комп'ютерної грамотності інженерів, спецкурси у вузах, велика кількість курсів підвищення кваліфікації, проведених провідними системними інтеграторами [31].

1.3.2. Програмовані логічні контролери

Тверді обмеження на вартість і величезна різноманітність цілей у мехатроніці привели до неможливості створення універсального ПЛК, як це, наприклад, відбулося з офісними комп'ютерами. Область мехатроніки висуває безліч завдань, відповідно до яких розвивається й ринок, що містить сотні несхожих один на одного контролерів, які відрізняються десятками параметрів. Кожний виробник випускає кілька типів ПЛК різної потужності й вартості, щоб збільшити прибуток за рахунок сегментування ринку.

Вибір оптимального для конкретного завдання контролера ґрунтується звичайно на відповідності функціональних характеристик контролера розв'язуваному завданню за умови мінімальної його вартості. Враховуються також інші важливі характеристики (температурний діапазон, надійність, бренд виробника, наявність сертифікатів тощо).

Незважаючи на величезну різноманітність контролерів, у їхньому розвитку помітні наступні загальні тенденції:

- ▶ зменшення габаритних розмірів;
- ▶ розширення функціональних можливостей;
- ▶ збільшення числа підтримуваних інтерфейсів і мереж;
- ▶ використання мов програмування стандарту МЕК 61131-3;
- ▶ зниження ціни.

Ще однією тенденцією є поява в контролерах ознак комп'ютера (наявність миші, клавіатури, монітора, можливості підключення жорсткого диска), а в комп'ютерах - ознак контролера (розширений температурний діапазон, електронний диск, захист від пилу й вологи, кріплення на Din-

рейку, наявність сторожового таймера, збільшена кількість комунікаційних портів, функції самотестування й діагностики, контроль цілісності прикладної програми). З'явилися комп'ютери в конструктивних виконаннях для жорстких умов експлуатації. Апаратні відмінності між комп'ютером і контролером поступово зникають. Основними відмінними ознаками контролера залишаються його призначення й наявність технологічної мови програмування.

Для класифікації величезної різноманітності існуючих у цей час контролерів розглянемо їхні істотні відмінності. Основним показником ПЛК є число каналів вводу-виводу. За цією ознакою ПЛК діляться на наступні групи:

- нано-ПЛК (менше 16 каналів);
- мікро-ПЛК (більше 16, до 100 каналів);
- середні (більше 100, до 500 каналів);
- великі (більше 500 каналів).

За розташуванням модулів вводу-виводу ПЛК бувають:

- 1) моноблочними, у яких пристрій вводу-виводу не може бути вилучений з контролера або замінений на інший. Конструктивно контролер являє собою єдине ціле із пристроями вводу-виводу (наприклад, одноплатний контролер). Моноблочний контролер може мати, наприклад, 16 каналів дискретного вводу й 8 каналів релейного виводу;
- 2) модульними, що складаються із загального „кошика”, у якому розташовуються модуль центрального процесора й змінні модулі вводу-виводу. Склад модулів вибираються користувачем залежно від розв'язуваного завдання. Типове число слотів для змінних модулів - від 8 до 32;
- 3) розподіленими (з віддаленими модулями вводу-виводу), у яких модулі вводу-виводу виконані в окремих корпусах. Вони з'єднуються з модулем контролера по мережі (звичайно на основі інтерфейсу RS-485)

і можуть бути розташовані на відстані до 1,2 км від процесорного модуля.

Часто перераховані конструктивні типи контролерів комбінуються, наприклад, моноблочний контролер може мати кілька з'ємних плат; моноблочний і модульний контролери можуть бути доповнені віддаленими модулями вводу-виводу, щоб збільшити загальну кількість каналів.

Багато контролерів мають набір змінних процесорних плат різної продуктивності. Це дозволяє розширити коло потенційних користувачів системи без зміни її внутрішньої „конструкції”.

За конструктивним виконанням й способом кріплення контролери діляться на:

- ▶ панельні (для монтажу на панель або дверцята шафи);
- ▶ для монтажу на Din-рейку усередині шафи;
- ▶ для кріплення на стіні;
- ▶ для монтажу на стійці;
- ▶ безкорпусні (звичайно одноплатні) для використання в спеціалізованих конструктивних виконаннях виробників обладнання.

За областю застосування контролери діляться на наступні типи:

- універсальні загальнопромислові;
- для керування роботами;
- для керування позиціонуванням і переміщенням;
- комунікаційні;
- ПІД-контролери;
- спеціалізовані.

За способом програмування контролери бувають:

- ▶ програмовані з лицьової панелі контролера;
- ▶ програмовані переносним програматором;
- ▶ програмовані за допомогою дисплея, миші й клавіатури;
- ▶ програмовані за допомогою персонального комп'ютера.

Контролери можуть містити модулі вводу-виводу або не містити їх. Прикладами контролерів без модулів вводу-виводу є комунікаційні контролери, які виконують функцію міжмережного шлюзу або контролери, що одержують дані від контролерів нижнього рівня ієрархії у мехатронних системах.

Програмування мікроконтролерів звичайно здійснюється мовою Асемблер або Сі, хоча існують компілятори для інших мов, наприклад, Фортран. Використовуються також вбудовані інтерпретатори Бейсика.

Найвідоміші компілятори Сі для мікроконтролерів: Codevisionavr (для AVR); IAR (для будь-яких мікроконтролерів), Winavr (для AVR і AVR32), Keil (для архітектури 8051 і ARM), Hitech (для архітектури 8051 і PIC від Microchip).

Для налагодження програм використовуються програмні симулятори (спеціальні програми для персональних комп'ютерів, що імітують роботу мікроконтролера), внутрішньоюсхемні емулятори (електронні пристрої, що імітують мікроконтролер, які можна підключити замість нього до розроблювального вбудованого пристрою) і інтерфейс JTAG [31].

1.3.3. Архітектура мікроконтролера

Архітектурою контролера називають набір його основних компонентів і зв'язків між ними. При проектуванні мікроконтролерів доводиться дотримувати балансу між розмірами й вартістю з однієї сторони й гнучкістю й продуктивністю з іншої. Для різних використань оптимальне співвідношення цих й інших параметрів може різнитись дуже сильно. Тому існує велика кількість типів мікроконтролерів, що відрізняються архітектурою процесорного модуля, розміром і типом вбудованої пам'яті, набором периферійних пристроїв, типом корпусу тощо.

Основні вимоги до мікроконтролерів, які входять у структуру мехатронної системи: низька вартість, висока надійність, високий ступінь мініатюризації, мале енергоспоживання, працездатність у жорстких умовах

експлуатації; достатня продуктивність для виконання всіх необхідних функцій.

Типовий ПЛК включає центральний процесор, пам'ять, мережеві інтерфейси й пристрої вводу-виводу (рис. 1.34). Іноді ця конфігурація доповнюється пристроєм для програмування й пультом оператора, пристроями індикації, рідше - принтером, клавіатурою і мишею.

Процесорний модуль містить у собі мікропроцесор (центральний процесорний пристрій - ЦПП), запам'ятовувальні пристрої, годинник реального часу й сторожовий таймер. Терміни „мікропроцесор” і „процесор” у цей час стали синонімами, оскільки всі процесори, виконуються у вигляді великих інтегральних мікросхем, тобто є мікропроцесорами.

Основними характеристиками мікропроцесора є розрядність (у ПЛК використовуються 8-, 16- і 32-розрядні мікропроцесори), тактова частота, архітектура, наявність операцій із плаваючою точкою, типи підтримуваних портів вводу-виводу, температурний діапазон працездатності й споживана потужність.

Продуктивність мікропроцесорів з однією і тією ж архітектурою пропорційна тактовій частоті. Більшість контролерів використовують мікропроцесори зі скороченим набором команд. Скорочений набір команд дозволяє будувати більш ефективні компілятори й конвеєр процесора, здатний за кожний такт видавати результат виконання чергової команди.

Для контролерів, що виконують інтенсивну математичну обробку даних, важлива наявність математичного співпроцесора (допоміжного процесора, що виконує операції з плаваючою точкою) або сигнальних процесорів, у яких операції типу $Y=AB+X$ виконуються за один такт. Сигнальні процесори дозволяють прискорити виконання операцій згортки або швидкого перетворення Фур'є.

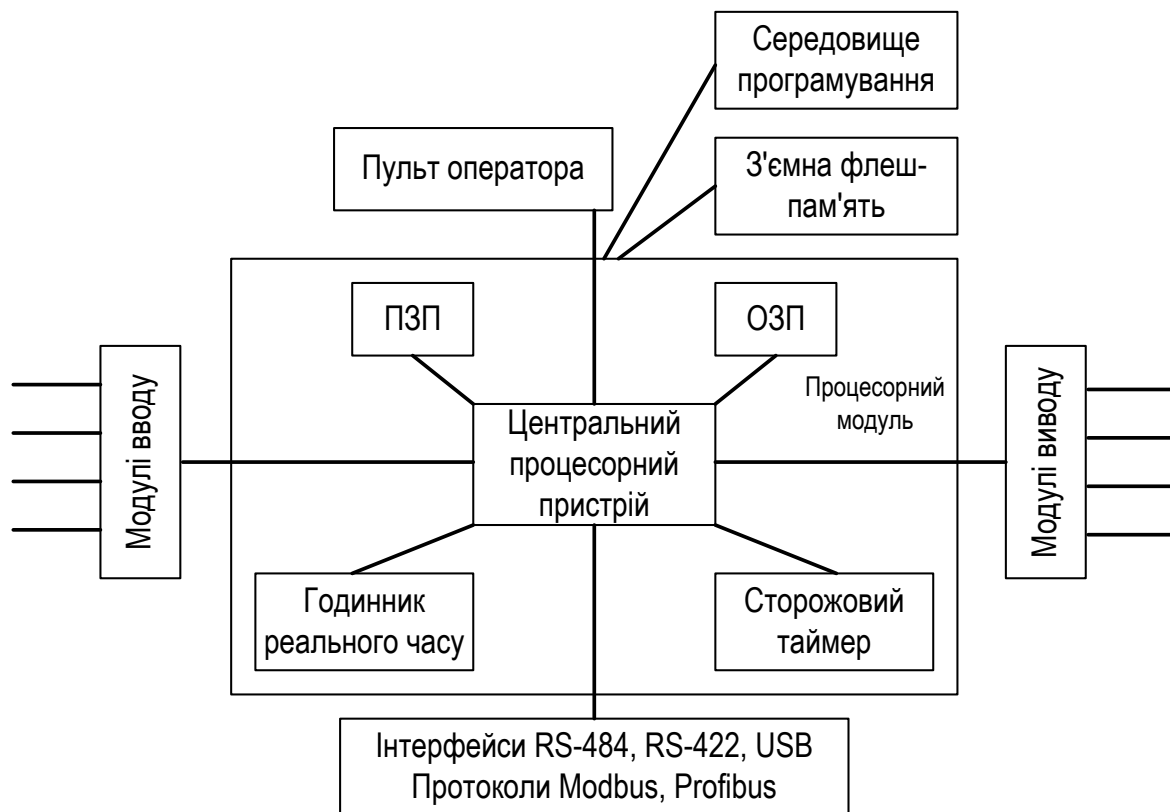


Рис. 1.34 Типова архітектура ПЛК

Об'єм пам'яті визначає кількість змінних (тегів), які можуть бути оброблені в процесі функціонування ПЛК. У мікропроцесорах час доступу до пам'яті є одним з істотних факторів, що обмежують їх швидкодію. Тому пам'ять ділять на кілька рівнів ієрархії залежно від частоти використання даних, які в ній зберігаються.

Ієрархія пам'яті відноситься до істотних характеристик архітектури процесора, оскільки вона дозволяє знизити негативний вплив повільної пам'яті на швидкодію мікропроцесора. Основними типами пам'яті є постійний запам'ятовувальний пристрій (ПЗП), оперативний запам'ятовувальний пристрій (ОЗП) і набір регістрів. Регістри є самими швидкодіючими елементами пам'яті, оскільки вони використовуються арифметико-логічним пристроєм (АЛП) для виконання елементарних команд процесора. ПЗП використовують для зберігання рідко змінюваної інформації, такої як операційна система, драйвери пристроїв, модуль програми користувача. ОЗП використовується для зберігання даних, які багаторазово

змінюються в процесі роботи контролера, наприклад, значення тегів, результати проміжних обчислень, діагностична інформація, масиви, виведені на графіки, дані для відображення на дисплеї.

У якості ПЗП (або ROM – Read Only Memory) звичайно використовується пам'ять, яку можна перепрограмувати (EEPROM – Electrically Erasable Programmable ROM). Різновидом EEPROM є флеш-пам'ять, принцип дії якої заснований на зберіганні заряду в конденсаторі, утвореному плаваючим затвором і підложкою МОП-транзистора. Особливістю флеш-пам'яті є її енергонезалежність, тобто в ній зберігаються дані навіть при виключеному живленні. Стирання й перезапис у флеш-пам'яті виконується не окремими комірками, а великими блоками, тому вона одержала назву, що походить від англійського „flesh” – „спалах”. Недоліком усіх ПЗП є низька швидкодія.

Кількість циклів запису інформації у флеш-пам'ять обмежена й становить кілька десятків тисяч разів. За конструктивним виконанням й інтерфейсам флеш-пам'ять поділяється на Compact Flash (CF), Memory Stick (MS), Secure Digital (SD), Multi Media Card (MMC), Smart Media Card (SMC), RS MMC та флеш-пам'ять, що може бути впаяна в плату або бути з'ємною.

У якості ОЗП сучасні мікропроцесори використовують статичну пам'ять (SRAM - Static Random Access Memory) і динамічну (DRAM – Dynamic Random Access Memory), SDRAM-пам'ять (Synchronous DRAM). SRAM виконується на тригерах, інформація в яких зберігається необмежено довго при наявності живлення. У динамічній пам'яті інформація зберігається на конденсаторах, і тому DRAM вимагає періодичної регенерації (перезарядження конденсаторів). До недоліків тригерної пам'яті відноситься її висока вартість, пов'язана з низькою щільністю компонування тригерів на кристалі, і мале відношення ємності до ціни. Перевагою є висока швидкодія, що досягає гігагерц, у той час як пам'ять на конденсаторах не може працювати на частотах вище сотень герц. Обидва типи пам'яті (DRAM і SRAM) не можуть зберігати інформацію при відключенні живлення ПЛК.

Тому деякі типи ПЛК використовують батарейне живлення пам'яті для збереження працездатності системи автоматизації після короткочасного переривання живлення.

Моноблочні й модульні контролери використовують, як правило, паралельну шину для обміну даними з модулями вводу-виводу, що дозволяє на порядок підвищити швидкість їх опитування в порівнянні з послідовною шиною. Послідовна шина контролера (на основі інтерфейсу RS-485) використовується для підключення до нього віддалених (розподілених) модулів вводу-виводу.

Програмування контролерів малої потужності виконується за допомогою кнопок, розташованих на лицьовій панелі або за допомогою переносного пульта для програмування. У якості пульта останнім часом використовується комп'ютер формату „ноутбук”. Програмування потужних контролерів виконується за допомогою персонального комп'ютера, на якому встановлюється спеціальне програмне забезпечення, наприклад CoDeSys, що виконує трансляцію технологічної мови стандарту MEK 61131-3 у код, який виконується процесором і завантажується в ПЗП ПЛК, наприклад, через порт Ethernet.

Сторожовий таймер (Watch Dog Timer - WDT) являє собою лічильник, який рахує імпульси тактового генератора й у нормальному режимі періодично скидається (перезапускається) працюючим процесором. Якщо процесор „зависає”, то сигнали скидання не надходять у лічильник, він продовжує лічити імпульси й при досягненні деякого порога виробляє сигнал „скидання” для перезапуску „завислого” процесора.

Годинник реального часу являє собою кварцевий годинник, який живиться від батарейки й тому продовжує працювати при виключеному ПЛК. Годинник реального часу використовується, наприклад, для керування вуличним освітленням залежно від часу доби, у системах охорони об'єктів і в інших випадках застосування, коли необхідна прив'язка даних або подій до астрономічного часу.

Процесорний модуль ПЛК виконує наступні завдання:

- ▶ збирає дані з модулів вводу в пам'ять й відсилає дані з пам'яті в модулі виводу;
- ▶ виконує обмін даними із пристроєм для програмування контролера;
- ▶ видає мітки годин реального часу;
- ▶ здійснює обмін даними із промисловою мережею;
- ▶ реалізує стек протоколів промислової мережі (для цієї мети можуть використовуватися допоміжні комунікаційні процесори);
- ▶ виконує початкове завантаження й виконання операційної системи;
- ▶ виконує завантажувальний модуль програми користувача системи автоматизації;
- ▶ керує актами обміну з пам'яттю.

Однією з тенденцій у розвитку ПЛК є використання процесорних модулів різної потужності для одного конструктиву контролера. Це дозволяє одержати серію контролерів різної потужності й тим самим покрити більший сегмент ринку, а також виконати модернізацію контролерів, куплених споживачами, шляхом заміни всього одного модуля. До основних характеристик процесорного модуля відносяться:

- тип операційної системи (Windows CE, Linux, DOS тощо);
- наявність виконавчого середовища для стандартної системи програмування на мовах MEK 61131-3;
- типи підтримуваних інтерфейсів (RS-232, RS-422, RS-485, CANopen, Ethernet);
- типи підтримуваних мереж (Modbus, Profibus, Ethernet, CANopen, DeviceNet тощо);
- можливість підключення пристроїв індикації або інтерфейсу оператора (світлодіодного або рідкокристалічного індикатора, клавіатури, миші, дисплея);
- розрядність (8, 16, 32 або 64 біта);
- тактова частота мікропроцесора й пам'яті;

- час виконання команд;
- об'єм, ієрархія й типи пам'яті (ОЗП, кеш, ПЗП-флеш, зйомна флеш тощо);
- типи вбудованих функцій (ПД-регулятор, лічильники, ШІМ, алгоритми позиціонування й керування рухом тощо);
- бренд виробника (Intel, AMD, Atmel, RealLab тощо).

Швидкодію процесорного модуля ПЛК звичайно оцінюють за часом виконання логічних команд, оскільки вони найпоширеніші при реалізації алгоритмів керування.

Величезна різноманітність завдань, які покладаються на ПЛК, і сильна залежність ціни від потужності контролера є причиною великої різноманітності використовуваних мікропроцесорів, від простих і дешевих 8-розрядних до самих високопродуктивних мікропроцесорів, включаючи двоядерні й чотирьохядерні процесори.

Восьмирозрядні мікропроцесори мають великий успіх в автономних ПД-контролерах і мікро-ПЛК для нескладного алгоритмічного керування верстатами, теплицями, невеликими технологічними апаратами, у якості міжмережних шлюзів. Їхньою перевагою є висока надійність, пов'язана із простотою програмного забезпечення.

Звичайно мікропроцесори, що використовуються в ПЛК, на кілька поколінь відстають від процесорів офісних персональних комп'ютерів (ПК) у зв'язку з відносно малим обсягом ринку ПЛК, який не забезпечує окупність розробки нового контролера за період зміни покоління мікропроцесорів.

Стандартними напругами живлення ПЛК є напруги 12, 24 і 48 В. Джерелом електричної енергії звичайно є промислова мережа (напруга 220 В, частота 50 Гц). У випадку розподілених систем автоматизації джерело живлення може бути розташоване на певній відстані від ПЛК, тому напруга на клеммах ПЛК або модулів вводу-виводу може сильно відрізнятись від напруги джерела живлення внаслідок спаду напруги на опорі кабелю живлення. Для розв'язання цієї проблеми кожний ПЛК або кожний модуль

віддаленого вводу забезпечується вбудованим стабілізатором напруги, який забезпечує нормальне їхнє функціонування в діапазоні напруг від 10 до 30 В.

Низька напруга живлення дозволяє жити контролери від акумуляторів бортових мереж транспортних засобів або від переносних акумуляторів. У ПЛК іноді використовують батарею для живлення годинника реального часу (які повинні функціонувати при виключеному ПЛК) і для збереження інформації в ПЗП на час аварійних перерв живлення.

Найпоширенішим представником сімейства мікроконтролерів є 8-розрядні прилади. Вони пройшли у своєму розвитку шлях від найпростіших приладів з відносно слаборозвиненою периферією до сучасних багатофункціональних контролерів, що забезпечують реалізацію складних алгоритмів керування в реальному масштабі часу. Причиною життєздатності 8-розрядних мікроконтролерів є використання їх для керування об'єктами де використовуються, в основному, алгоритми з перевагою логічних операцій, швидкість обробки яких практично не залежить від розрядності процесора.

Росту популярності 8-розрядних мікроконтролерів сприяє постійне розширення номенклатури виробів, що випускаються такими відомими фірмами, як Motorola, Microchip, Intel, Zilog, Atmel і багатьма іншими. Сучасні 8-розрядні МК мають, як правило, ряд характерних ознак. Перелічимо основні з них:

- 1) модульна організація, при якій на базі одного процесорного ядра (центрального процесора) проектується ряд (лінійка) мікроконтролерів, які відрізняються об'ємом і типом пам'яті програм, об'ємом пам'яті даних, набором периферійних модулів, частотою синхронізації;
- 2) використання закритої архітектури мікроконтролера, яка характеризується відсутністю ліній магістралей адреси й даних на виводах корпусу мікроконтролера. Таким чином, мікроконтролер являє собою закінчену систему обробки даних, нарощування можливостей якої не передбачається;

3) використання типових функціональних периферійних модулів (таймери, процесори подій, контролери послідовних інтерфейсів, аналого-цифрові перетворювачі тощо) незначні відмінності в алгоритмах роботи в мікроконтролерів різних виробників.

При модульному принципі побудови всі мікроконтролери одного сімейства містять процесорне ядро, однакове для всіх мікроконтролерів даного сімейства, і змінний функціональний блок, який відрізняє мікроконтролери різних моделей [31].

1.3.4. Характеристики мікроконтролерів

Продуктивність ПЛК оцінюється за наступними параметрами:

- ▶ тривалість контролерного циклу (період зчитування значень із каналів вводу, обробки в процесорі й запису в канали виводу);
- ▶ час виконання команд (окремо логічних, з фіксованою й із плаваючою точкою);
- ▶ пропускна здатність шини між контролером і модулями вводу-виводу;
- ▶ пропускна здатність промислової мережі;
- ▶ час циклу опитування всіх контролерів в одномастерній мережі або цикл обігу маркера для багатомастерних мереж з маркером.

Контролер у системах автоматизації виконує циклічний алгоритм, що включає введення даних і розміщення їх в ОЗП, обробку даних і виведення.

Тривалість контролерного циклу (його ще називають робочим циклом) залежить від кількості модулів вводу-виводу й наявності в них замаскованих (виключених із процедури обміну) входів-виходів, розраховується для кожної конфігурації мехатронної системи окремо. При інтенсивній математичній обробці даних (наприклад, при цифровій фільтрації, інтерполяції або ідентифікації об'єкта керування в режимі нормального функціонування системи) тривалість контролерного циклу суттєво залежить від швидкодії процесорного модуля. У контролерний цикл входить також обслуговування апаратних ресурсів ПЛК (забезпечення роботи системних

таймерів, оперативне самотестування, індикація стану), контроль часу циклу, мережевий обмін, керування багатозадачністю, відображення процесу виконання програми на дисплеї тощо.

Перед початком роботи ПЛК виконує завантаження операційної системи й програми користувача, початкове тестування апаратури. ПЛК звичайно може працювати в режимі налагодження, покрокового виконання програми, перегляду й редагування значень змінних тощо.

У процесі функціонування ПЛК дані, уведені із пристроїв вводу, „розташовуються” в ОЗП й протягом робочого циклу контролера не змінюються. Пряме читання входу під час виконання одного циклу не виконується. Це прискорює процес обробки даних і виключає непередбачені ситуації.

У системах з розподіленими по об'єкту модулями вводу-виводу тривалість контролерного циклу може визначатися пропускнуою здатністю промислової мережі, що в ряді випадків є обмеженням на граничну кількість модулів вводу-виводу.

Вимоги до тривалості контролерного циклу суттєво залежать від області застосування ПЛК. При керуванні тепловими процесами тривалість циклу може становити одиниці й десятки секунд, у завданнях для керування верстатами вона вимірюється мілісекундами, при опитуванні датчиків температури на елеваторі контролерний цикл вимірюється годинами.

Час реакції контролера - це інтервал часу від моменту появи впливу на систему (з боку модулів вводу або оператора) до моменту „вироблення” (розрахунку) відповідної реакції. Час реакції контролера залежить від тривалості робочого циклу контролера, який визначається швидкодією модулів вводу-виводу й продуктивністю процесора.

У контролерах для відповідальних використань можуть бути передбачені наступні функції самодіагностики: виявлення помилок центрального процесора; сигналізація про спрацьовування сторожового таймера; виявлення відмови батареї або джерела живлення; виявлення збою

пам'яті; перевірка програми користувача; виявлення виходу з ладу запобіжника; виявлення обриву або короткого замикання у колі датчика й навантаження.

У контролерах для систем протиаварійного захисту і сигналізації, а також для небезпечних промислових об'єктів може бути передбачена можливість резервування окремих частин системи: промислової мережі, процесорного модуля або контролера, джерела живлення, мережевого сервера, замкнених контурів автоматичного регулювання, модулів вводу-виводу. Об'єктом резервування звичайно є найбільш відповідальна або найбільш ненадійна частина системи. Можливість гарячої заміни елементів системи (тобто без відключення живлення) досягається одночасно апаратними й програмними засобами. Апаратно передбачається незалежність початкового стану пристрою від черговості подачі сигналів на його клеми в процесі заміни; програмно забезпечується можливість тимчасової відсутності компонента системи без її зависання або переходу в аварійні режими.

Надійність контролерів характеризується наробітком на відмову, яка визначається відношенням сумарного часу працездатного стану контролера до математичного очікування числа його відмов протягом цього часу або наробітком до відмови - часом від початку експлуатації до першої відмови. Надійність пов'язана із припустимими механічними перевантаженнями - амплітудою вібрації в необхідному діапазоні частот, припустимим прискоренням при ударі.

Для підвищення безпеки систем автоматизації в контролерах використовуються команди для установки початкового стану виходів відразу після подачі живлення або в аварійному режимі. Ці стани вибираються таким чином, щоб після відновлення напруги живлення при випадковому його перериванні або в аварійному режимі системи виконавчі пристрої перебували в безпечному для персоналу або системи стані. Наприклад, у системах з нагріванням безпечним буде стан відключеного нагрівача, у піднімальних механізмах - стан гальмування. Наявність команд керування безпечними

станами дозволяє реалізувати операцію автоматичного рестарту мехатронної системи після переривання живлення або після відновлення працездатного стану. Здатність контролера переводити свої виходи в заздалегідь певний стан відразу після виявлення зниження напруги живлення або після внутрішньої відмови називається відмовостійким відключенням.

Якщо повторний запуск ПЛК виконується після того, як усі динамічні дані (змінні входів-виходів, стан внутрішніх регістрів, таймерів, лічильників, програмні контексти) були повернуті в заздалегідь певний стан, то такий запуск називається холодним рестартом. Холодний рестарт може бути як ручним, так і автоматичним.

Гарячим рестартом називають повторний запуск ПЛК, який виконується настільки швидко після провалу живлення, що всі динамічні змінні не встигають змінитися й тому працездатність відновлюється таким чином, що живлення „неначе й не пропадало”.

Завадостійкість контролера звичайно оцінюється за його відповідністю комплексу стандартів за електромагнітною сумісністю.

Промислові контролери використовують гальванічну ізоляцію для усунення паразитних зв'язків по загальному провіднику, землі й для захисту обладнання від високих напруг.

Ступінь захисту від впливу навколишнього середовища позначається буквами „IP”, за якими ідуть дві цифри. Перша цифра означає ступінь захисту виробу від попадання всередину твердих сторонніх тіл, друга цифра означає ступінь захисту виробу від попадання води. Розшифрування позначень наведено в табл. 1.4.

Важливим параметром ПЛК є час від замовлення до поставки й наявність декількох незалежних постачальників. Великі строки поставки пояснюються як далекістю виробника від споживача, так і прагненням виробників і постачальників мінімізувати внутрішні витрати за рахунок зменшення складських запасів.

Таблиця 1.4. Розшифровка позначення захисту виробу від впливу навколишнього середовища

Перша цифра	Ступінь захисту	Друга цифра	Ступінь захисту
0	Захист відсутній	0	Захист відсутній
1	Захист від твердих тіл розміром більше 50 мм	1	Захист від крапель води
2	Захист від твердих тіл розміром більше 12 мм	2	Захист від крапель води при ухилі до 15 ⁰
3	Захист від твердих тіл розміром більше 2,5 мм	3	Захист від дощу
4	Захист від твердих тіл розміром більше 1 мм	4	Захист від бризок
5	Захист від пилу	5	Захист від водяних струменів
6	Пилонепроникність	6	Захист від хвиль
		7	Захист при зануренні у воду
		8	Захист при тривалому зануренні у воду

Якість документації, наявність групи технічної підтримки, швидкість реакції на запити, наявність курсів для навчання споживачів відіграють також важливу роль при виборі контролера з величезного їхнього різноманіття [31].

1.3.5. Пристрої збору даних

Системи збору даних у даний час є загальнодоступним засобом одержання експериментальної інформації, що у першу чергу пов'язано із широким поширенням персональних комп'ютерів. Системи збору даних знаходять використання для наукових досліджень, керування виробничими процесами, моніторингу в промисловості, медицині, метеорології, космонавтиці й інших областях людської діяльності. Автоматизований збір даних дозволяє одержати дані нової якості, які неможливо одержати іншими засобами. Це результати статистичної обробки величезного числа вимірів, отриманих у цифровій формі, можливість реєстрації випадкових подій з

недосяжною раніше точністю за часом і за величиною, реєстрація швидкозмінних процесів. Завдяки різкому здешевленню систем збору даних у порівнянні з вартістю людської праці з'явилася велика кількість областей застосування, де раніше використовувалася ручна реєстрація даних: у теплицях, елеваторах, на метеостанціях, у процесі приймально-здавальних і сертифікаційних випробувань продукції, на складах, у промислових холодильниках, при автоматизації наукового експерименту тощо.

Основною відмінністю систем збору даних від ПЛК є відсутність у них алгоритму керування, тобто відсутність необхідності в потужному контролері й мові МЕК 61131-3, а також наявність великого об'єму пам'яті для ведення архіву. Хоча системи збору даних можна побудувати на будь-якому ПЛК, але у зв'язку із зазначеними вище особливостями вони займають окремий сегмент ринку і їх виділяють в окрему групу засобів автоматизації.

Системи збору даних можуть застосовуватися в реальному часі, наприклад, для моніторингу (спостереження) різних процесів, ідентифікації аварійних ситуацій у технологічних процесах, а також можуть застосовуватися для архівування даних, коли їх обробка відділена від процесу збору невизначеним інтервалом часу. У системах реального часу поточні дані зберігаються протягом деякого заданого часу в кільцевому буфері. Дані, що застаріли, витісняються новими. В архівуючих системах використовуються накопичувачі інформації великої ємності й дані обробляються після завершення збору.

Архівуючі системи збору даних (логгери, самописці) можуть бути автономними пристроями, побудованими на основі мікроконтролера (наприклад, бортові самописці літаків, електронні лічильники тепла або електроенергії, портативні електрокардіографи). Дані, зібрані логгерами, для обробки переносяться в комп'ютер за допомогою, наприклад, USB-флеш-пам'яті або через послідовний інтерфейс.

Системи збору даних, побудовані на основі комп'ютера, звичайно є стаціонарними й використовують універсальне програмне забезпечення, таке

як MATLAB, LabVIEW, MS Excel [31], яке дозволяє не тільки зібрати дані, але й обробити їх.

Для реєстрації швидкоплинних процесів (з необхідною частотою дискретизації більше 1 МГц) використовуються системи з паралельною шиною, у тому числі плати для шини PCI комп'ютера. Комп'ютерні плати мають обмежену кількість входів, що визначається комп'ютерним конструктивом, і вимагають зовнішніх клемних блоків для приєднання джерел сигналу, створюючи незручності при монтажі системи.

Для реєстрації повільних процесів зручніше використовувати зовнішні пристрої, які підключаються до комп'ютера через COM, USB або Ethernet-порт. Зовнішні пристрої відрізняються також меншим рівнем шумів, у той час як плати, що вставляються в комп'ютер, піддаються впливу електромагнітних шумів від електричних (цифрових) кіл комп'ютера.

Система збору даних може бути розподіленою, коли пристрій вводу розподілений територіально по об'єкту збору даних, а отримані дані сходяться до єдиного накопичувача й оброблювача даних за допомогою мережевих технологій. Мережеві (розподілені) системи збору даних мають властивість практично необмеженого збільшення (нарощування) числа каналів, однак вони мають обмеження на швидкість передачі даних в мережі.

Для типових завдань збору даних промисловістю випускаються пристрої з невеликим числом входів (від декількох десятків до декількох сотень). Для великих систем (від одиниць до сотень тисяч входів) розроблюються спеціалізовані системи. До них можна, наприклад, віднести систему „COMPASS”, що збирає дані з 250 тисяч датчиків у ядерному центрі CERN зі швидкістю 160 Мбіт/с.

Входи систем збору даних можуть бути універсальними (потенціальними й струмовими) або спеціалізованими (наприклад, для термопар, для термоперетворювачів опору або для тензодатчиків). Системи зі спеціалізованими входами економічно більш ефективні для споживача. Універсальні входи використовуються разом з вимірювальними

перетворювачами фізичних величин у струм або напругу. Існують також системи з гібридними входами, наприклад, коли кілька входів приймають сигнали термопар, інші входи - сигнали тензодатчиків, треті - сигнали термометрів опору тощо.

Входи можуть бути диференціальними, одиночними, цифровими або дискретними (двійковими). Диференціальні входи дозволяють більш ефективно пригнічувати зовнішні перешкоди, що наводяться у кабелі, якій передає сигнал від датчика до модуля введення. Для передачі сигналу найчастіше використовується напруга в діапазоні $0 \dots \pm 5$ В, $0 \dots \pm 10$ В або струм $0 \dots 20$ мА, $4 \dots 20$ мА. Сигнали напруги виробляються джерелами напруги й мають високу завадостійкість до ємнісних наведень, сигнали струму виробляються джерелами струму й стійкі до індуктивних наведень. Дискретні входи приймають логічні сигнали („0” або „1”), які надходять від кінцевих вимикачів, датчиків охоронної або пожежної сигналізації, електромагнітних реле, датчиків наявності напруги тощо. Цифрові входи приймають сигнали від пристроїв із цифровим виходом, наприклад, від цифрових датчиків температури.

Основними параметрами систем збору даних є кількість каналів, похибка, динамічна похибка, смуга пропускання, роздільна здатність, ефективне число розрядів, частота дискретизації, наявність гальванічної ізоляції входів та інтерфейсу, наявність захистів від недбалого використання, перевантажень і перегріву.

Системи збору даних звичайно мають 4, 8, 16, 32 або 64 входів, які опитуються по черзі або одночасно. Системи з одночасним опитуванням складаються з ідентичних каналів, які виконують аналого-цифрове перетворення вхідної величини паралельно, тобто одночасно для всіх каналів. Такі системи зустрічаються рідко через високу вартість. Звичайно опитування входів виконується по черзі, за допомогою комутатора. Тому дані різних каналів виявляються зсунутими за часом на деяку затримку, рівну відношенню періоду опитування до кількості каналів.

Прикладом системи збору даних може служити система RealLab!, побудована за модульним принципом. Таку систему з необхідною кількістю входів можна зібрати з модулів - окремих „будівельних” блоків. Модулі з’єднуються між собою за допомогою промислового інтерфейсу RS-485 і розміщуються або в загальній монтажній шафі, або розподілені по об’єкту збору даних таким чином, щоб зменшити довжину кабелю від датчика до модуля. Зібрані дані в цифровій формі передаються по промисловій мережі в центральний комп’ютер або контролер. Модулі RealLab! можуть працювати в стандартних мережах Modbus. Кожний модуль у мережі має свою адресу, тому для опитування модулів комп’ютер посилає їм команду, що містить адресу й код операції, яку необхідно виконати.

Наближення модулів введення до датчиків має кілька переваг. По-перше, скорочується кількість проводів, оскільки цифровий інтерфейс RS-485 має тільки два провідника, а передає дані від великої кількості модулів. Це зручно при зборі даних з територіально розподілених об’єктів, наприклад при зборі даних про температуру й вологість в теплиці, яка має площу 10 Га, у багатоповерховому будинку або на елеваторі. По-друге, знижується потужність перешкод завдяки скороченню довжини провідників з аналоговими сигналами, спрощується технічне обслуговування й діагностика системи.

Завдяки застосуванню стандартного протоколу обміну в системі збору даних на модулях RealLab! можуть бути включені пристрої введення інших виробників, наприклад, лічильник електроенергії або метеостанція.

1.3.6. Комп’ютер у мехатронних системах

Незважаючи на існуючі традиції застосування ПЛК для вирішення проблем у мехатроніці, багато завдань набагато ефективніше вирішуються за допомогою персональних комп’ютерів. Комп’ютер може бути використаний також як пульт оператора (диспетчера) або виконувати одночасно функції

контролера й пульта оператора. У завданнях мехатроніки широко застосовують як офісні, так і промислові комп'ютери [31].

1.3.6.1. Комп'ютер як контролер

Комп'ютер може бути перетворений у повноцінний контролер, якщо на нього встановити:

- ▶ систему програмування контролерів (наприклад, CoDeSys);
- ▶ електронний диск замість звичайного жорсткого диска;
- ▶ плати вводу-виводу або зовнішні модулі вводу-виводу;
- ▶ сторожовий таймер.

Електронний диск являє собою ПЗП, що складається з мікросхем флеш-пам'яті й конструктивно виконаний у вигляді корпуса звичайного жорсткого диска формату 2,5", 3,5" або 5,25" зі стандартними інтерфейсами IDE або SCSI. Основними перевагами електронного диска є його висока стійкість до вібрацій, відсутність акустичного шуму, висока надійність, низьке енергоспоживання, мала вага.

Плати й зовнішні модулі вводу-виводу для комп'ютера випускаються багатьма незалежними виробниками, що стало можливим завдяки наявності стандартів на конструктивне виконання й шини як промислових, так і офісних комп'ютерів.

Сторожовий таймер служить для перезавантаження комп'ютера у випадку його зависання. Відмітимо, що комп'ютер має набагато більший, аніж контролер, час перезавантаження при використанні офісної операційної системи Windows.

До переваг персональних комп'ютерів (ПК) при їхньому використанні в якості контролерів відносяться:

- ▶ на порядок більший об'єм оперативної пам'яті;
- ▶ наявність жорсткого диска із практично необмеженим об'ємом;
- ▶ на порядки більш висока швидкодія;

- ▶ швидкий розвиток апаратного забезпечення (модернізація комп'ютерів відбувається кожні 4...6 місяців, контролерів - через 3...5 роки);
- ▶ у кілька разів нижча ціна;
- ▶ наявність великої кількості фахівців, що добре володіють персональним комп'ютером;
- ▶ підтримка комп'ютером DVD і CD-ROM дисководів, принтера, звукових карт, систем розпізнавання й синтезу мови, систем розпізнавання образів і тексту, а також інших засобів, недоступних для контролера;
- ▶ легка заміна комп'ютера на більш потужний без зміни програмного забезпечення;
- ▶ підтримка клавіатури, миші й монітора;
- ▶ можливість комплектації комп'ютера із частин, вироблених великим числом незалежних виробників;
- ▶ можливість створення всієї системи автоматизації за допомогою SCADA-пакетів (наприклад, Master SCADA), без залучення систем програмування контролерів і засобів зв'язку контролера з комп'ютером;
- ▶ наявність стандартів на шини, що є причиною великої кількості незалежних постачальників стандартних комп'ютерних плат.

Хоча промислові комп'ютери, наближаючись за умовами експлуатації до контролерів, втрачають деякі з переваг ПК (швидкодію, швидкість модернізації, низьку ціну), інші переваги ПК роблять їх широко розповсюдженими й необхідними в промисловості.

З розвитком промислових комп'ютерів і контролерів відбувається стирання відмінностей між ними. Проте залишається ряд відмінностей, які вже не можна назвати істотними, але які зустрічаються найбільш часто. Типовий комп'ютер, на відміну від типового контролера, має:

- мишу, клавіатуру й монітор;
- жорсткий диск;

- вузький температурний діапазон (від 0 до +50 °С);
- операційну систему Windows;
- вентилятор для охолодження процесора й блоку живлення.

У промислових комп'ютерах іноді використовують операційну систему реального часу, однак вона не завжди є необхідною навіть у системах автоматичного керування. Недоліком комп'ютерів з ОС Windows є великий час перезавантаження системи при її зависанні [31].

1.3.6.2. Комп'ютер для спілкування з оператором

Найбільш типовим застосуванням ПК у завданнях автоматизації є організація робочого місця оператора (диспетчера). Комп'ютер тут виконує роль людино-машинного інтерфейсу. Для поліпшення ергономічності й ефективності роботи оператора використовують SCADA-пакети з використанням звуку, анімації, високоякісної кольорової графіки й безлічі інтелектуальних функцій, що полегшують роботу оператора.

Для створення людино-машинного інтерфейсу використовують монітори із сенсорним екраном, плазмові панелі, проектори екрана комп'ютера на стіну, трекбол, звукові колонки, сирени, клавіатури зі ступенем захисту від IP-20 до IP-67.

Сенсорний екран являє собою скляну або пластикову пластину, прикріплену поверх екрана звичайного монітора. Для визначення координат торкання екрана пальцем оператора використовують в основному чотири фізичні принципи: поглинання пальцем енергії поверхневих акустичних хвиль, зміна ємнісного струму при торканні екрана, зміна опору між двома прозорими провідними поверхнями при натисненні пальцем і принцип затінення інфрачервоних променів, що виходять із випромінювачів по периметру екрана. Найбільше поширення одержали резистивні та ємнісні сенсорні екрани.

У промисловій автоматизації використовуються також панелі оператора, які замість клавіатури мають кілька спеціалізованих кнопок, а

замість монітора - мініатюрний дисплей, на якому поміщається всього кілька рядків тексту або графіки. Такі пристрої підключаються до СОМ-порту комп'ютера.

Для монтажу на панель (на дверцята шафи, на стіну) використовують панельні комп'ютери, які поєднують в одному конструктиві процесорний блок, дисплей із сенсорним екраном і клавіатуру [31].

1.3.6.3. Промислові комп'ютери

Промислові комп'ютери суттєво відрізняються від офісних за конструктивними ознаками, однак використовують ті ж процесори та архітектуру. Основні відмінності є такі:

- ▶ роз'єми для з'ємних плат встановлюються на пасивній об'єднаній панелі, а не на материнській платі;
- ▶ для з'ємних плат використовуються надійний штирьовий роз'єм;
- ▶ для зміни плат не потрібно розкривати корпус;
- ▶ використовуються спеціалізовані промислові комп'ютерні шини CompactPCI, COMExpress, VME та інші;
- ▶ замість жорсткого диска може бути використана флеш-пам'ять;
- ▶ наявність сторожового таймера;
- ▶ застосування вентиляторів зі змінним фільтром або відсутність вентиляторів;
- ▶ міцна несуча конструкція з надійним кріпленням плат пружними планками із гвинтовими затискачами;
- ▶ застосування блоків живлення підвищеної надійності, із захистом від коротких замикань по виходу й з розширеним діапазоном мережних напруг (від 100 до 250 В);
- ▶ резервування блоків живлення;
- ▶ наявність енергонезалежної оперативної пам'яті (з акумуляторним живленням), яка зберігає дані при збоях або зникненні живлення;

- ▶ в одному конструктиві й на одній об'єднаній шині може розташовуватися кілька комп'ютерних систем;
- ▶ наявність з'ємної флеш-пам'яті;
- ▶ застосування, крім Windows, операційних систем DOS, Linux.

Конструктивне виконання промислового комп'ютера забезпечує захист від пилу, вологи, вібрації, електромагнітних впливів і полегшує технічне обслуговування. Для роботи в діапазоні температур від -40 до +70 °C використовується індустриальна елементна база з розширеним температурним діапазоном або, якщо це можливо, штучний підігрів і примусове охолодження.

Монтаж промислового комп'ютера може бути виконаний у стандартній 19-дюймовій стійці, на панелі, на столі, на стіні, на Din-рейці й у спеціалізованому конструктиві. Органи керування промисловим комп'ютером (кнопка скидання, живлення й клавіатура) можуть закриватися дверцятами із ключем.

За швидкодією і об'ємом пам'яті промислові комп'ютери завжди уступають офісним, що пов'язане з великою тривалістю розробки нових модифікацій: за час розробки промислового комп'ютера встигає змінитися кілька поколінь офісних ПК. Це пояснюється незрівнянно меншим об'ємом ринку промислових комп'ютерів у порівнянні з офісними, що визначає розмір інвестицій у нові проекти.

Переважає більшість промислових комп'ютерів відносяться до магістрально-модульних системам. Вони не мають об'єднаної материнської плати, замість неї використовується пасивна крос-плата (сполучна плата), що містить тільки роз'єми, з'єднані загальною магістраллю. Усі плати комп'ютера вставляються в корпус попереду й закріплюються планками з амортизаторами для підвищення вібростійкості. Це зроблене для зручності заміни плат при обслуговуванні. Плати з'єднуються із крос-платою за допомогою штирьових з'єднань, які набагато надійніше з'єднань в офісних ПК. Число слотів для з'ємних плат перевищує 10 [31].

1.3.7. Пристрої вводу-виводу

Пристрої (модулі) вводу-виводу є інтерфейсом між процесором ПЛК і „реальним світом”. Бажано мати в процесорі значення вимірюваних сигналів у будь-який момент часу. Однак, оскільки кількість каналів вводу-виводу в деяких системах може досягати тисяч, а вимірювальні канали завжди мають обмежену пропускну здатність, вимірювані значення надходять у процесор у дискретні моменти часу.

Існує кілька рівнів і способів опитування великої кількості каналів вводу. Сучасний модуль вводу має свій власний мікроконтролер, який виконує циклічне опитування всіх своїх каналів і поміщає отримані дані в буфер. Якщо по алгоритму роботи мехатронної системи використовуються тільки кілька каналів модуля, то канали, що не використовуються можна замаскувати (виключити їх із процедури опитування), якщо це потрібно для збільшення швидкодії системи. При надходженні в модуль команди зчитування значень із входів зібрані дані передаються з буфера модуля в ПЛК, де надходять в буфер сервера або в певну область ОЗП.

Опитування модулів може виконуватися циклічно з однаковою частотою для всіх модулів або з різною частотою. Другий варіант дозволяє зменшити завантаженість шини, по якій виконується обмін даними між модулями вводу й процесорним модулем.

Циклічне опитування всіх модулів із заздалегідь заданою частотою сильно завантажує шину, по якій модулі вводу зв'язуються із процесором. Це особливо очевидно, якщо процесор сканує входи для виявлення сигналу від аварійного датчика, який може спрацювати один раз в 10 років, або якщо вводяться дані від датчика температури в умовах, коли температура постійна. У подібних випадках більш ефективні багатомастерні шини (наприклад, CAN), які дозволяють використовувати режим підписки, при якому процесор модуля вводу, у якому відбулася зміна стану входу, є ініціатором обміну даними.

Найбільше поширення мехатроніці знайшли одномастерні шини й циклічне опитування модулів вводу в силу своєї простоти й порівняно низької вартості.

Модулі вводу й виводу в промисловій автоматизації мають гальванічну ізоляцію між вхідними (вихідними) затискачами й шиною контролера. Допустима напруга прикладена до ізоляції становить від 2500 (рідше від 500) до 4000 В.

Іноді потрібно виконати одночасне опитування входів усіх модулів вводу або вивести дані одночасно в канали всіх модулів виводу. Для вирішення цієї проблеми використовують ширококомвні команди, які сприймаються всіма модулями одночасно й вони виконують введення або виведення даних у свої буферні реєстри одночасно. Після цього звичайним циклічним опитуванням дані по черзі вводяться в процесорний модуль.

Модулі введення з'єднуються із процесором послідовною або паралельною шиною. У магістрально-модульних системах використовуються паралельні шини ISA, PCI, VME, у модульних ПЛК - нестандартні послідовні й паралельні шини. У контролерах з розподіленими (віддаленими) модулями вводу-виводу найпоширеніші послідовні шини на основі інтерфейсів RS-485 і CAN.

Перевагою паралельної шини є висока пропускна здатність, що дозволяє виконувати сканування модулів вводу з високою частотою й використовувати модулі аналогового вводу з тактовою частотою АЦП до 100 кГц. Однак невелика довжина паралельної шини, обмежена розсинхронізацією окремих біт у переданому слові, не дозволяє підключити до одного контролера більше 32 модулів. Контролери з послідовною шиною мають протилежні властивості. Крім того, вони дозволяють організувати розподілені системи збору даних.

Більшість паралельних і послідовних шин контролерів є одномастерними, оскільки багатомастерні шини суттєво складніші й дорожчі.

Обмін даними з модулем виконується за адресою, яка звичайно записується в ПЗП модуля. Іноді адресою є номер слота, у який вставляється модуль або положення мікроперемикача.

Електричні кола входів і виходів модулів вводу повинні мати гальванічну ізоляцію. Гальванічна ізоляція може бути поканальна (кожний канал ізольований від інших), або групова. Звичайно використовується групова ізоляція. У віддалених модулях розподілених ПЛК може бути використана індивідуальна гальванічна ізоляція інтерфейсу RS-485 кожного модуля або групова ізоляція інтерфейсів декількох модулів за допомогою одного модуля (повторювача інтерфейсів). Для передачі напруги живлення в ізольовану частину модуля використовуються DC-DC перетворювачі.

Сучасні модулі вводу-виводу можуть виконувати крім функцій вводу деяку обробку інформації, яка вводиться, й додаткові функції: компенсацію температури холодного спаю термопар, лінеаризацію нелінійних датчиків, діагностику обриву датчика, автоматичне калібрування, ПД-регулювання, керування рухом. Перенос частини функцій контролера в модулі вводу-виводу є сучасною тенденцією, спрямованою на збільшення ступеня „розпаралелювання” завдань керування, забезпечення незалежності локальних модулів (які за своїми функціями наближаються до ПЛК) і зменшення потоку інформації між паралельно працюючими процесорами в модулях вводу-виводу. По такому принципу побудовані, зокрема, модулі RealLab! [31].

1.4. Питання та завдання для самоперевірки

1. Як класифікують датчики за принципом дії?
2. Вкажіть основні характеристики датчиків.
3. Який принцип дії потенціометричних датчиків?
4. За якою схемою вмикаються тензометричні датчики?
5. Які фізичні величини можна виміряти за допомогою датчика Холла?
6. Що таке енкодер та який принцип його дії?
7. Які типи електромагнітних датчиків вам відомі?
8. В чому полягає різниця між аналоговою та дискретною формою представлення даних?
9. Які складові входять у структуру „інтелектуальних” датчиків?
10. Назвіть основні функції „інтелектуальних” датчиків.
11. Які вимоги пред’являють до приводів мехатронних систем?
12. З’ясуйте структуру електричного приводу, який входить у мехатронну систему.
13. В чому полягає принцип виникнення електромагнітного моменту у двигуні постійного струму?
14. Що таке кроковий електродвигун та в яких сферах мехатроніки його використовують?
15. Які види частотного керування асинхронних двигунів вам відомі?
16. Вкажіть структуру частотного перетворювача та принцип його дії.
17. Які основні функції мають сучасні частотні перетворювачі?
18. Що таке інтелектуальний мехатронний модуль руху?
19. Які загальні тенденції проявляються у розвитку мікроконтролерних пристроїв.
20. Які складові входять у типову структуру логічного мікроконтролера?
21. Вкажіть основні характеристики мікроконтролерів.
22. Приведіть приклад пристрою збору даних.
23. Які основні застосування комп’ютера у сучасних мехатронних системах?

2. МЕТОДИ КЕРУВАННЯ МЕХАТРОННИМИ СИСТЕМАМИ

2.1. Ієрархія систем керування у мехатроніці

Структура модулів систем керування припускає багаторівневу організацію й включає стратегічний, тактичний і виконавчий рівні керування, що мають доступ до сенсорної інформації для розв'язування завдань керування даного рівня.

Слід підкреслити, що принцип ієрархічної побудови систем керування складними динамічними об'єктами сам по собі не є оригінальним. Він був сформульований ще на рубежі сімдесятих-восьмидесятих років ХХ століття стосовно проблем керування багатофункціональними роботами. Нова концепція ієрархічної побудови систем керування визначає необхідність інтелектуалізації кожного з рівнів керування. Розглянемо завдання, що розв'язуються на кожному рівні керування.

Стратегічний рівень керування призначений для планування руху мехатронної системи в умовах неповної інформації про зовнішнє середовище й об'єкт керування. Планування рухів означає розбивку завдання руху на послідовність погоджених у часі дій, оперативну корекцію руху з урахуванням зміни середовища й формалізацію цілей керування для кожного із цих дій. Формалізація цілей керування означає, що для кожної дії повинні бути записані математичні співвідношення, виконання яких забезпечує успішне виконання кожної дії. Сенсори стратегічного рівня повинні відповідати органам почуттів людини (технічний зір, тактильне й силомоментне відчуття, пристрій аналізу звукових і ультразвукових сигналів тощо).

Тактичний рівень виконує перетворення команд керування рухом, що надходять зі стратегічного рівня керування в програму керування, яка визначає закони погодженого руху в часі усіх ланок механічного пристрою мехатронної

системи з врахуванням технічних характеристик блоку приводів (у першу чергу обмежень на узагальнені швидкості, прискорення й сили).

На *виконавчому рівні* виконується розрахунки й видача керуючих сигналів на блок приводів мехатронної системи відповідно до програми керування й з урахуванням технічних характеристик силових перетворювачів.

Для ієрархічних систем керування в мехатроніці діє принцип, сформульований в 1989 р. проф. Сарідісом (*Saridis G.N.*), згідно з яким в міру просування від вищих до нижчих рівнів керування знижується інтелектуальність системи, але підвищується її точність (даний принцип в англійській літературі має абревіатуру IPDI – *Increasing Precision with Decreasing Intelligence*) [32]. При цьому під „інтелектуальністю” розуміється здатність системи здобувати спеціальні знання, що дозволяють уточнити поставлене завдання і визначити шляхи його розв’язку, а під „неточністю” – невизначеність в операціях по вирішенню даного завдання.

Датчики тактичного й виконавчого рівнів повинні забезпечувати достовірну інформацію про стан системи й об’єкта керування, а також адекватність перетвореної інформації до цілей функціонування мехатронної системи.

При керуванні складними динамічними об’єктами в реальних умовах виникає невизначеність, пов’язана з формуванням керуючих впливів по вимірюваній та апріорній інформації.

Аналіз джерел невизначеності дозволяє розділити невизначеність на два види: *передбачувану* й *непередбачувану*.

Передбачувана невизначеність має місце в тому випадку, коли відома апріорна інформація про характер закону зміни керованих координат. Наприклад, траєкторії руху робочого органа робота, рівняння руху якого відомі, можна апроксимувати кінцевим набором сплайнів; при цьому невизначеність

буде полягати тільки у виборі відповідного сплайна. А *непередбачувана невизначеність* має місце, коли такої інформації немає.

Так, наприклад, для складальних роботів, які працюють у рамках певного набору технологічних рухів, характерних для складального роботизованого процесу, невизначеність може бути класифікована як передбачувана. Роботи, призначені для екстремальних середовищ, в основному працюють в умовах непередбачуваної невизначеності, тому що траєкторію їх руху не можна заздалегідь передбачити.

Такий підхід дозволив виділити два класи інтелектуальних систем керування, що принципово відрізняються по архітектурі, принципу дії і області застосування. Інтелектуальні системи керування I роду призначені для роботи в умовах передбачуваної невизначеності, в основному пов'язаної із взаємним впливом окремих ступенів вільності складного багатоланкового об'єкта керування один на одного. Інтелектуальні системи керування II роду орієнтовані на роботу в умовах непередбачуваної невизначеності й функціонують на основі реалізації своїх адаптивних властивостей за рахунок використання технології асоціативної пам'яті.

Системи керування в умовах невизначеності, як правило, будуються за ієрархічним принципом. На рис. 2.1 представлена ієрархічна схема інтелектуальної системи керування I роду, що забезпечує роботу в умовах передбачуваної невизначеності.

На стратегічному рівні керування формуються траєкторія й динамічні характеристики руху по ній. Функції інтелектуального регулятора розподіляються на тактичному й виконавчому рівнях. На тактичному рівні керування шляхом розв'язку зворотної задачі кінематики формуються програмні узагальнені координати. У результаті розв'язку зворотної задачі динаміки (по відомій моделі об'єкта керування й апріорній інформації про залежність структури й параметрів об'єкта від траєкторії руху) визначаються

структура й параметри регулятора, які реалізуються на виконавчому рівні й можуть мінятися в процесі функціонування.

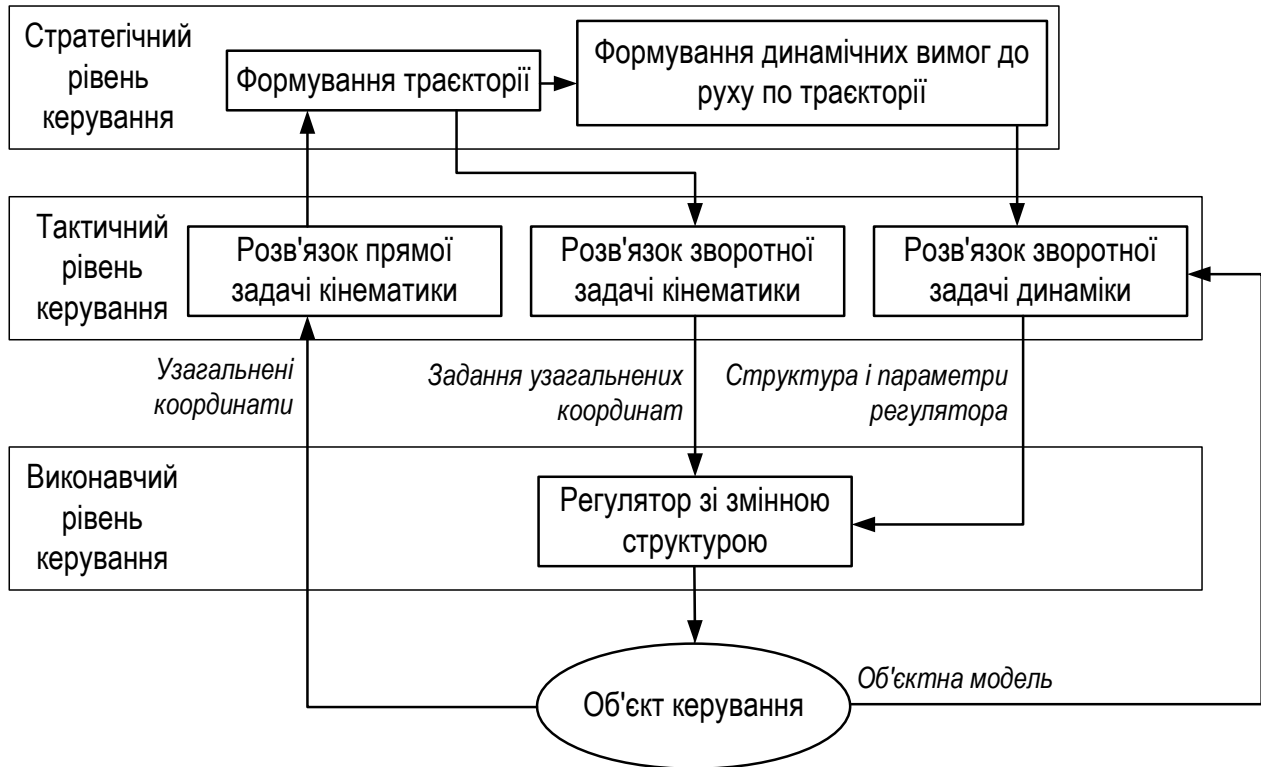


Рис. 2.1. Ієрархічна схема інтелектуальної системи керування I роду

Корекція програмної траєкторії проводиться по реальній траєкторії, обчисленій за допомогою розв'язку прямої задачі кінематики для реальних значень узагальнених координат. Оскільки в запропонованому підході в окремих приводах контролюються тільки власні узагальнені координати, то врахування зміни динамічних властивостей об'єкта фактично зводиться до „розв'язки” приводів, тобто до незалежності роботи приводів один від одного.

Для забезпечення „розв'язки” приводів в умовах обмеженої потужності обчислювальних засобів пропонується, використовуючи знання про специфіку виконуваного автоматизованого технологічного процесу, кінематичну схему й динамічні характеристики виробничого устаткування, здійснювати рух робочого органа по спеціальним траєкторіям. Ці траєкторії, названі *технологічними рухами*, вибираються таким чином, щоб, з одного боку, вони

забезпечували заданий технологічний процес, а з іншого – задовольняли найбільш простим рівнянням динаміки (що дозволяє розв’язувати їх у реальному масштабі часу на основі обчислення швидкостей і прискорень окремих приводів, здійснюючи тим самим „розв’язку”). При такому підході необхідність в обміні інформацією між окремими приводами повністю відпадає.

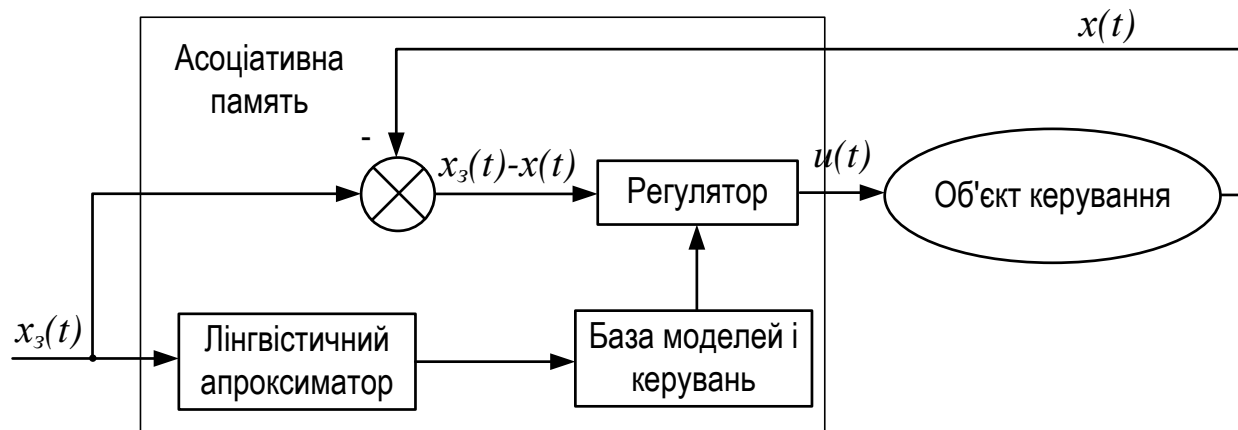


Рис .2.2. Функціональна схема інтелектуальної системи керування I роду

На рис. 2.2 наведена функціональна схема інтелектуальної системи керування I роду, у якій пристрій, названий *лінгвістичним апроксиматором* (ЛА), замінює траєкторію руху робочого органа робота набором спеціальних траєкторій (сплайнів). За допомогою лінгвістичного апроксиматора з *бази моделей і керувань* (БМК) вибираються відповідні налаштування регулятора, структура й параметри якого визначаються типом апроксимуючого технологічного руху робочого органа й залежать від вимірюваних швидкостей і прискорень тільки даного привода.

Робота інтелектуальної системи керування I роду здійснюється на основі знань, що утримуються в ЛА й БМК і які базуються на:

- аналізі автоматизованого технологічного процесу;
- аналізі кінематичних схем і рівнянь динаміки виробничого обладнання;
- теорії автоматичного керування й зворотних задач динаміки.

Враховуючи складність алгоритмів керування, що реалізують взаємозв'язність ступенів вільності в об'єкті, для забезпечення необхідної швидкодії системи ЛА, БМК й регулятор будуються на базі асоціативної пам'яті. Адаптація до змінних параметрів в структурі об'єкта за кожним ступенем вільності відбувається на основі прогнозування поведінки інших ступенів вільності і їх впливу на даний ступінь.

На рис. 2.3 представлена функціональна схема інтелектуальної системи керування II роду адаптації в умовах непередбачуваної невизначеності.

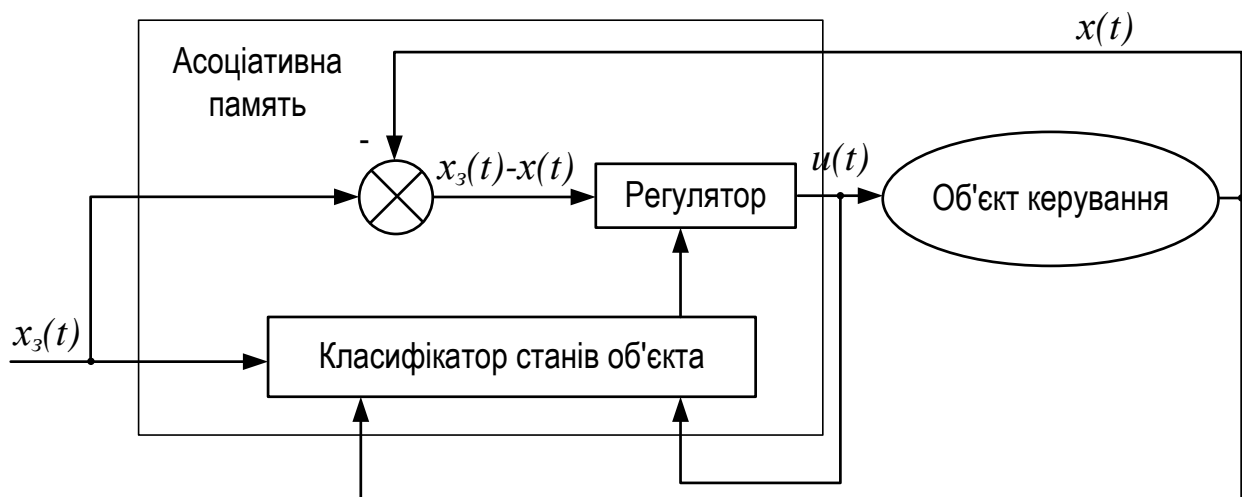


Рис. 2.3. Функціональна схема інтелектуальної системи керування для роботи в умовах непередбачуваної невизначеності

Класифікатор стану об'єкта керування й формувач керувань (у цьому випадку мова йде про формування керуючих впливів безпосередньо на об'єкт керування) реалізуються в асоціативній пам'яті, у якій на підставі інформації про вхідні й вихідні сигнали об'єкта керування відбуваються такі дії:

- проводиться класифікація стану об'єкта;
- формується вектор параметрів, які налаштовуються, і керуючий вплив, який відповідає йому й вхідному завданню.

Таким чином, робота в умовах непередбачуваної невизначеності, коли не відомо, що у наступний момент часу відбудеться із складним динамічним об'єктом, приводить до того, що будувати й аналізувати математичні моделі, що відбивають специфіку взаємодії й взаємного впливу окремих його частин, не має сенсу через їхню складність і практичну неадекватність реальному об'єкту.

Неадекватність моделей, як правило, викликана похибками вимірів і обчислень, спрощенням моделей і відмінністю параметрів реальних систем від модельних. Це приводить до того, що в ієрархічній системі керування (рис. 2.4), що забезпечує ефективну роботу в умовах непередбачуваної невизначеності, врахування динамічних властивостей об'єкта керування повинно відбуватися на виконавчому рівні, який у свою чергу повинен гарантувати виконання завдань тактичного рівня на необхідних швидкостях.

Така побудова системи керування приводить до того, що на приводному рівні складні динамічні об'єкти керування вважаються зосередженими й однозв'язними зі змінними динамічними параметрами й збуреннями, приведеними до валів виконавчих двигунів.

На тактичному рівні повинні вирішуватися тільки кінематичні завдання, які відбивають статичні властивості об'єкта. Відповідно до необхідної траєкторії виконавчого органа, швидкостей та прискорень руху в глобальній системі координат на тактичному рівні обчислюються задані швидкості й прискорення узагальнених координат, а на приводному рівні регулятор з адаптивним налаштуванням параметрів забезпечує їхнє відпрацьовування.

Якість виконання завдань контролюється обрахованими координатами реального стану об'єкта і при необхідності проводиться корегування траєкторії. Треба мати на увазі, що кожний рівень керування може мати різний ступінь інтелектуальності [33].

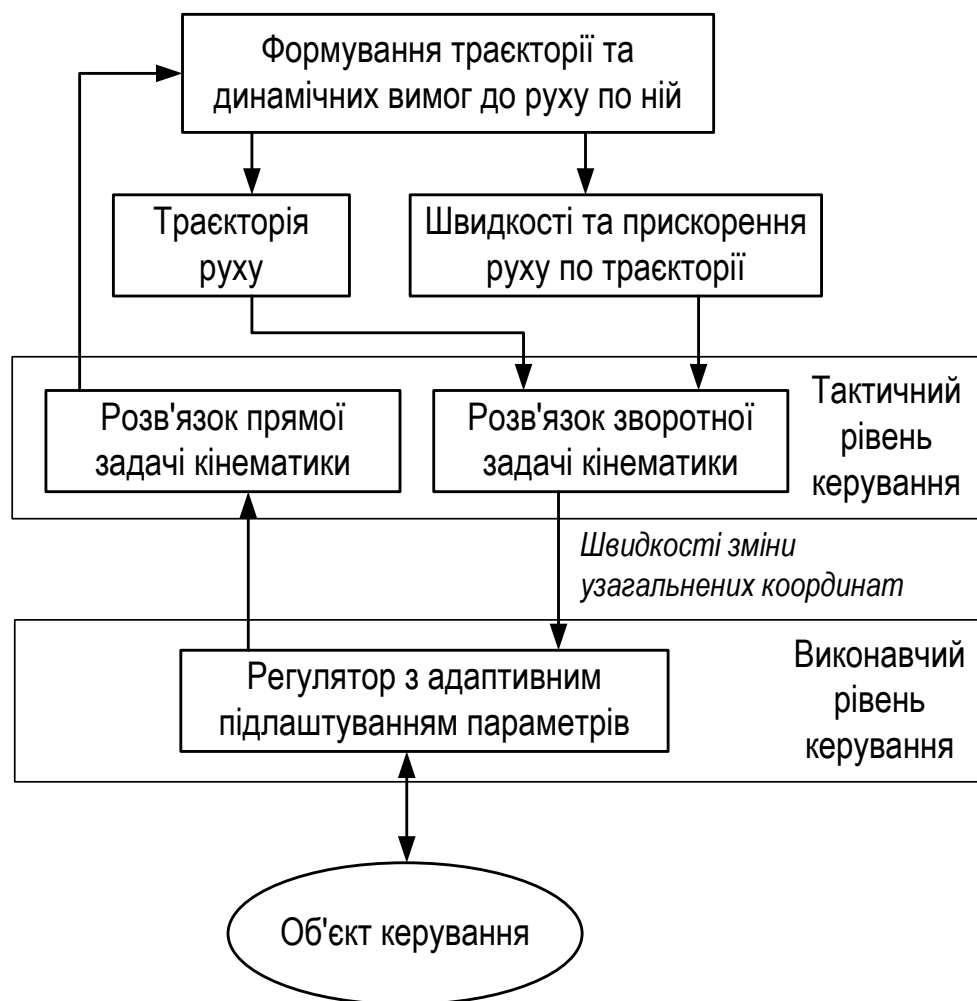


Рис. 2.4. Ієрархічна схема інтелектуальної системи керування II роду

2.2. Проектування систем керування мехатронними об'єктами

Застосування теорії керування завжди закінчується керуванням реальним об'єктом. Вивчення об'єкта керування є невід'ємною частиною процесу конструювання системи керування. Цьому є природні пояснення:

- ▶ по-перше, планування керуючих впливів (конструювання регулятора) завжди враховує поведінку об'єкта керування, хоча б у деякому наближенні;
- ▶ по-друге, як і багато інших технічних об'єктів, система керування й принципи її роботи повинні бути протестовані перед застосуванням у реальних умовах.

Ідеальним варіантом є ситуація, коли об'єкт керування постійно доступний і над ним можуть бути проведені будь-які експерименти. У дійсності, це можливо виконати лише у виняткових випадках. Звичайно ми маємо лише обмежений доступ до об'єкта керування, і число експериментів, які можна над ним провести, також обмежене. Поведінку об'єкта керування можна (з деяким наближенням) простежити за допомогою комп'ютера – за математичною моделлю. Будь-яка модель – це деякий спрощений і обмежений опис реального об'єкта. Модель може мати дуже глибоку деталізацію, але майже завжди реальні процеси в об'єкті складніші, чим в їх моделях. Моделювання в теорії керування відіграє виняткову роль. До моделей необхідно ставитися як до „молодших братів” реального об'єкта керування: усі вони корисні й починають роботу звичайно з найпростіших моделей, поступово ускладнюючи їх при необхідності, і пам'ятаючи про те, що реальний об'єкт – це краща модель. Чи корисні прості моделі складних явищ? Безумовно, корисні, якщо вони дозволяють передбачати поведінку реального об'єкта (може бути й дуже грубо). Чи потрібно ускладнювати моделі? Це необхідно виконати, якщо прості нас не влаштовують. Тут виникає поняття адекватності моделі – це чисто суб'єктивний критерій: якщо модель забезпечує прийнятну для нас точність, то вважаємо модель придатною й адекватною. У теорії керування моделювання й експериментування становлять єдиний процес проектування. Це пов'язано з тим, що нас не цікавить як така точність моделювання об'єкта керування, нас цікавить якість самого керування. Якщо проста модель дозволяє досягти прийнятної якості керування, то не має сенсу переходити до більш складних моделей, і можна вважати на цьому процес проектування закінченим. Але якість керування можливо визначити тільки після моделювання й тестування об'єкта керування разом з регулятором – такі експерименти називають тестування із замкненим зворотним зв'язком (*closedloop response test*). Таким чином, адекватність моделі для систем керування визначається прийнятністю

поведінки побудованого керуючого реальним об'єктом регулятора, який отриманий на основі розглянутої моделі.

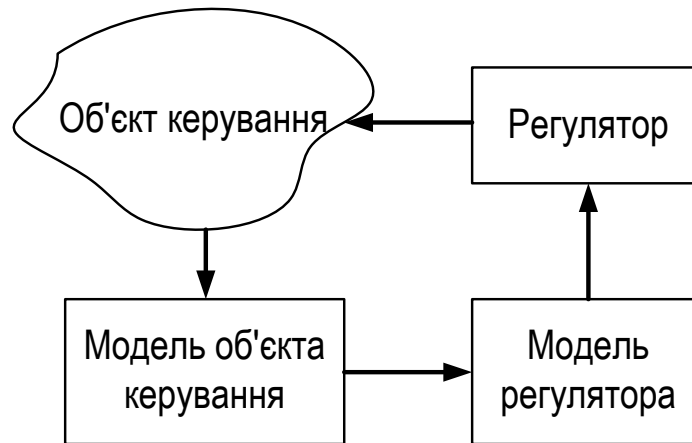


Рис. 2.5. Етапи проектування системи керування

На практиці звичайно виконують таке проектування систем керування мехатронними об'єктами (рис. 2.5):

1. **Побудова моделі об'єкта керування.** Модель об'єкта керування будується або з аналізу фізичних процесів, що відбуваються в ньому, або на основі комбінування моделей процесів, які виникають у різних частинах об'єкта керування. Ще один шлях – використовувати які-небудь стандартні абстрактні моделі, які „підганяються” до результатів вимірів, отриманих при експериментуванні з реальним об'єктом керування. Звичайно поєднують всі ці способи, використовуючи подекуди інтуїцію й природні припущення. Побудова моделі – це справжнє мистецтво (і як у будь-якому мистецтві тут важливо вчасно зупинитися).
2. **Побудова моделі системи керування.** Отже, знання про об'єкт керування акумульовані в його моделі. Аналізуючи цю модель, можна побудувати регулятор для вирішення завдань керування. Регулятор може мати стандартну структуру, а може, навпаки, бути досить специфічним для керуваного об'єкта. У будь-якому разі, перш ніж застосовувати регулятор до реального об'єкта керування, завжди починають з моделювання

замкненої системи (об'єкта керування й регулятора). Отже, предмет теорії керування – це вивчення моделі об'єкта керування й побудова моделі регулятора, який вирішує задані завдання керування. Існує величезний арсенал формальних і „напівформальних” методів і інструментів, за допомогою яких можна одержати (синтезувати) регулятор за заданою моделлю об'єкта керування.

3. **Реалізація системи керування.** Модель регулятора – це опис того, як він працює. На сучасному технічному рівні, регулятори реалізуються за допомогою спеціальних комп'ютерних систем. Завдання реалізації регулятора полягає в переході від його моделі до формального опису алгоритму на якій-небудь мові програмування для обраної комп'ютерної системи. Крім того, повинні враховуватися багато технічних особливостей, наприклад: як реалізовано вимірювання стану об'єкта керування і як подаються керуючі впливи на керований об'єкт. Цим моментам часто мало приділяють увагу.
4. **Тестування системи керування з реальним об'єктом.** Це завжди ключовий етап, який визначає наскільки врахована вся необхідна інформація про об'єкт керування, чи коректно реалізований сам регулятор. Випробування з реальними об'єктами керування завжди пов'язані із певним ризиком. Звичайно в системах керування передбачають можливість безумовної ручної зупинки й вимикання на випадок виникнення небезпечних ситуацій. Крім того, необхідно заздалегідь продумати програму тестування й випробувань, з якої було б ясно наскільки система керування задовольняє поставленим завданням. Якщо за результатами випробувань, система керування не задовольняє якимось необхідним їй властивостям, то це означає, що цикл проектування необхідно повторити. Виходячи із ситуації, або уточнюють модель (переходять до п.1), або вибирають інший метод для реалізації

моделі регулятора (переходять п.2), або розглядають питання про використання іншого обладнання (переходять до п.3).

Таким чином, проектування систем керування – це циклічний процес, який містить у собі експериментування, розрахунки і моделювання [34].

2.3. Основні поняття про будову та алгоритми роботи автоматичних систем керування мехатронними об'єктами

2.3.1. Основні поняття теорії керування

Поняття керування відомо нам із самого дитинства. Це одне з найбільш інтуїтивних і природних понять людської діяльності.

Сила й складність теорії керування полягає в тому, що вона розглядає явища в самих різних галузях науки й техніки, без якоїсь конкретної прив'язки. Системи керування – це, у першу чергу, інформаційні системи. Це означає, що має місце абстрагування від фізичної сутності процесів, що відбуваються в керованих системах, і описується їхній стан за допомогою інформаційних сигналів $x(t)$ – деяких змінних, значення яких змінюються з часом. Значення конкретного сигналу в системі керування може відповідати якомусь конкретному фізичному параметру (наприклад, швидкості руху), але в загальному випадку явної відповідності може й не бути. У цьому полягає одна з відмінних рис теорії керування, яка відрізняє її від інших дисциплін (наприклад, фізики), де принципово важлива фізична сутність змінних і залежностей між ними.

Керування – це досягнення заданого стану фізичного об'єкта за рахунок здійснення над ним певних дій. Фізичний об'єкт, стан якого необхідно змінити (наприклад, перемістити в інше положення, нагріти, розігнати до деякої швидкості тощо), називається об'єктом керування або в англійській літературі – *plant*. Об'єкт керування – це деяка система, яка має вихідні сигнали $y(t)$ і

вхідні $u(t)$. Значення вхідних сигналів завжди задаються ззовні, значення вихідних визначаються поведінкою об'єкта керування за вхідними сигналами. Прийнята наступна математична нотація: $u(t) \rightarrow y(t)$, це означає, що кожному вхідному сигналу ставиться у відповідність вихідний сигнал.

Більш конкретно, керування – це планування керуючого впливу і його застосування до об'єкта керування для досягнення заданого стану або режиму роботи об'єкта керування. Пристрій або підсистема, яка обчислює (виробляє) керуючий вплив на основі заданої мети та інформації про стан об'єкта керування, називається системою керування (в англійській літературі – *controller*).

В теорії керування прийняте, що об'єкт керування задається апіорі, тобто він не може змінюватись. Безумовно, у практиці конструювання мехатронних систем часто буває, що об'єкт керування й система керування „взаємодіють”: обирається конкретний об'єкт керування і вирішуються певні завдання керування. Якщо виникають принципові труднощі, то змінюють щось в об'єкті керування й знову займаються системою керування і так далі. Однак, майже всі методи теорії керування сконцентровані навколо заданого і незмінного об'єкта керування.

Регулятор завжди з'єднується послідовно з об'єктом керування в тому розумінні, що вихід регулятора – це керуючий вплив, який безпосередньо подається на об'єкт керування. Існують різні варіанти мети керування, але всі вони можуть розглядатися як окремий випадок однієї мети: намагаються досягти на виході об'єкта керування бажаного сигналу $r(t)$, тобто зробити так, щоб $r(t) \approx y(t)$ або в ідеалі $r(t) = y(t)$, де $r(t)$ – сигнал, заданий користувачем системи (об'єкта керування й регулятора в цілому) [34].

2.3.2. Стійкість динамічної системи

Користувача мехатронної системи мало цікавить як „всередині” працює система керування. Його завжди цікавить кінцевий результат – наскільки добре система керування справляється з покладеними на неї функціями. Будь-яка мета керування – трансформувати вихідний об’єкт керування, поведінка якого в певному сенсі нас не влаштовує, у деякий новий об’єкт із прийнятними для нас властивостями.

Центральним поняттям теорії керування є стійкість. У першому наближенні, поведінку будь-якої системи можна розділити на два типи:

- на обмежений вхідний вплив $u(t)$ система реагує обмеженою за величиною вихідною реакцією $y(t)$. Це означає, що енергія в системі не накопичується, а вся розсіюється з часом. Говорять, що в цьому випадку поведінка системи є стійкою (*stable*). Наприклад, на рис. 2.6 показано, що подача на вхід системи постійного керування призводить до того, що з часом вихідна змінна приймає усталене значення;
- на обмежений вхідний вплив $u(t)$ система реагує необмеженою за величиною вихідною реакцією $y(t)$. У цьому випадку, енергія в системі накопичується, що приводить до постійного зростання вихідних сигналів. Говорять, що в цьому випадку поведінка системи нестійка (*unstable*). Типовий вплив і приклад реакції нестійкої системи показано на рис. 2.7.

Звичайно, стійкість – це властивість самої системи. Більш строго, стійкість – це властивість поведінки системи: в одних режимах роботи система може бути стійкою, в інших – ні. У теорії керування існує багато інших понять стійкості, обґрунтованих теоретичними викладками, але на наш погляд, наведена вище класифікація досить проста для розуміння, тому будемо притримуватись її.

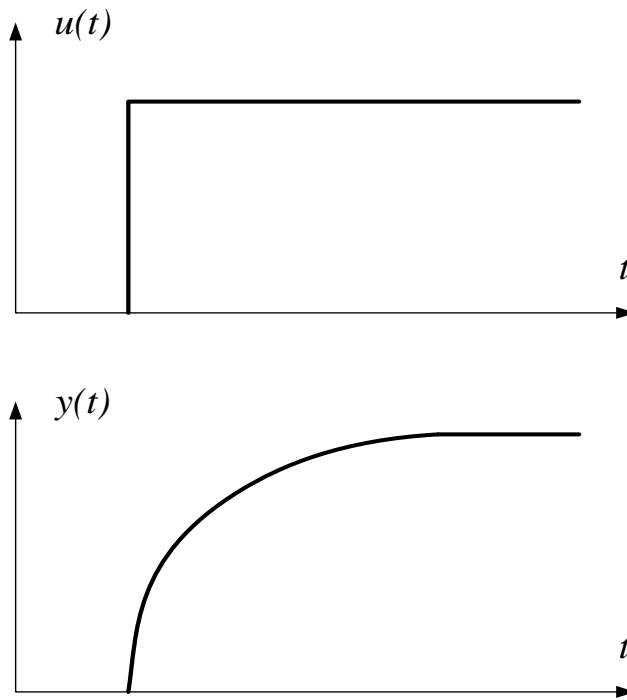


Рис. 2.6. Реакція стійкої системи на постійний за величиною вплив

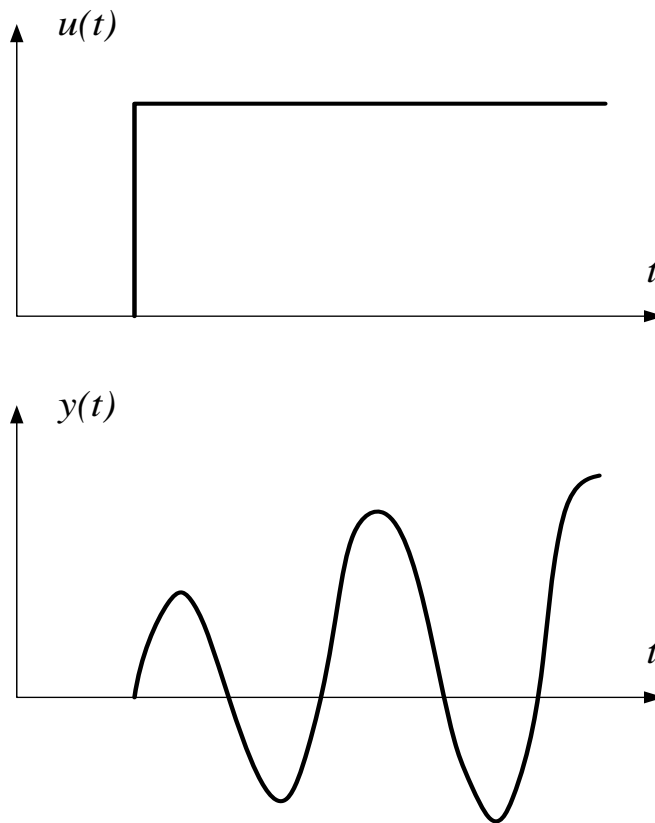


Рис. 2.7. Реакція нестійкої системи на постійний за величиною вплив

Властивість стійкості системи автоматичного керування ілюструють станами рівноваги кулі, що перебуває на різних поверхнях (рис. 2.8).



Рис. 2.8. Ілюстрація поняття стійкості системи: а) – *система стійка* (куля повертається в початкове положення, після закінчення дії сили, що змістила її із цього положення); б) – *система нестійка* (після дії сили куля не повертається у початкове положення)

Втрата стійкості виражається в неконтрольованому зростанні вихідних сигналів, що може спричинити катастрофічні наслідки при роботі реального об'єкта керування. Одна із найважливіших задач, які ставляться перед методами теорії керування – зробити з нестійкого об'єкта керування стійкий.

Існує багато критеріїв стійкості об'єкта залежно від типів його моделей. Для лінійних об'єктів застосовують два підходи до визначення стійкості: прямий (безпосередній), непрямий. При прямому підході записують будь-який процес керування й, знаючи його бажаний стан й виходячи з визначення стійкості, роблять висновки про його стійкість. Однак прямий метод дослідження стійкості об'єкта не завжди доцільний, а іноді й неможливий. Це буває у випадку, коли користувач працює тільки з його математичною моделлю, у вигляді, наприклад, диференціальних рівнянь. У цій ситуації для дослідження стійкості об'єкта використовуються звичайно наступні непрямі методи: алгебраїчні, кореневий, частотний.

Кількісна оцінка стійкості систем проводиться за допомогою системи показників, що характеризують запас стійкості. Запас стійкості - це кількісна характеристика ступеня віддалення системи від межі стійкості.

Забезпечення запасу стійкості необхідно з наступних причин:

- ▶ при складанні рівнянь, що описують окремі елементи об'єкта керування допускається деяка їх ідеалізація (враховуються тільки головні фактори й відкидаються другорядні);
- ▶ лінеаризація нелінійних рівнянь приводить до ще більшої наближеності (грубості, неточності) моделі об'єкта керування;
- ▶ конструктивні параметри елементів, що входять з коефіцієнти рівнянь, визначаються з деякою похибкою;
- ▶ при експлуатації систем можливі зміни параметрів елементів внаслідок температурних коливань, старіння, нестабільності тощо.

Наявність певного запасу стійкості гарантує збереження стійкості системи при зміні її параметрів у певних межах. Чим більший запас стійкості, тим менша ймовірність того, що система в процесі експлуатації стане нестійкою. Запас стійкості необхідний ще й тому, що він визначає характер перехідних процесів у системі. Наявність певного запасу стійкості забезпечує роботу реальної системи в області стійкості з необхідною якістю перехідного процесу [34].

2.3.3. Класифікація завдань автоматичного керування

Керування може здійснюватися з різними цілями. Залежно від цілей керування розрізняють наступні його види:

- стабілізація – керування з метою підтримки заданого постійного значення керованої величини об'єкта керування. Наприклад, підтримка постійної швидкості обертання електродвигуна;
- програмне керування – забезпечення заданої зміни в часі керованої величини. Керована величина в цьому випадку повинна змінюватися за відомою функцією часу. Наприклад, керування крутним моментом на валу двигуна у функції часу для розгону ланки робота;

- відстеження – керування з метою зміни керованої величини таким чином, щоб вона відтворювала б зміни деякої вимірюваної величини, закон зміни якої заздалегідь невідомий. Наприклад, повторення маніпулятором рухів людини у режимі on-line;
- оптимальне керування – керування, при якому переведення об’єкта керування з довільного вихідного стану в заданий стан здійснюється оптимально відповідно до обраного критерію оптимальності (наприклад, за мінімальний час або з мінімальними витратами енергії). Приклад: переміщення вантажу вантажопідйомним краном із одного заданого місця у інше за найкоротший час;
- екстремальне керування – автоматичне забезпечення екстремального значення вихідної величини об’єкта керування у змінних умовах його функціонування. Приклад: керування підводним роботом при дії на нього течії з якнайменшими витратами енергії;
- адаптивне керування – автоматична зміна характеристик керування таким чином, щоб об’єкт керування функціонував би щонайкраще при змінних умовах функціонування та змінних характеристиках самого об’єкта керування. Приклад: керування роботом, який виконує складні завдання розпізнавання об’єктів, їх захват різними захватними пристроями та інші маніпуляції над ними.

Залежно від того, чи враховується при керуванні об’єктом поточний стан об’єкта, керування може бути: розімкнутим, замкненим.

При розімкненому керуванні керуючий вплив формується пристроєм керування без врахування фактичного значення керованої величини. Керуючий вплив визначається на основі мети керування й відомих характеристик об’єкта. Таке керування інколи називають „жорстким”. Розімкнуте керування може застосовуватися для стабілізації й програмного керування. Розімкнута система керування застосовується в тому випадку, коли властивості об’єкта повністю

відомі, усі зовнішні впливи на об'єкт контролюються, і їхній вплив може бути зведений до нуля.

При замкненому керуванні керуючий вплив формується в безпосередній залежності від керованої величини. Керована величина в цьому випадку постійно контролюється й у системі керування є зворотний зв'язок з виходу об'єкта керування на вхід пристрою керування. Замкнена система здійснює керування на основі виміру похибки (різниці) між заданою змінною керованою величиною та її дійсним значенням.

Залежно від величин, що враховуються при визначенні керуючого впливу керування може бути:

- ▶ за відхиленням (за похибкою);
- ▶ за збуренням;
- ▶ комбіноване керування.

При керуванні за відхиленням пристрій керування контролює керовану величину $y(t)$ об'єкта керування й порівнює її із заданим значенням (уставкою) $r(t)$, розраховуючи їх відхилення:

$$\tilde{\alpha}(t) = y(t) - r(t). \quad (2.1)$$

За величиною відхилення (похибкою) визначається необхідний для її усунення керуючий вплив:

$$u(t) = A\{x(t)\}, \quad (2.2)$$

де A – оператор (деяка функція, закон), визначений вибраним законом керування.

При керуванні за збуренням передбачається відомим зв'язок між збуреннями, що діють на об'єкт і його вихідною величиною. Пристрій

керування контролює збурення й за його величиною визначає керуючий вплив на об'єкт керування, необхідний для компенсації цього збурення:

$$u(t) = A\{g(t)\}, \quad (2.3)$$

де $g(t)$ – функція збурення.

При керуванні за збуренням пристрій керування може компенсувати дію тільки тих збурень, які контролюються. Можливості такого керування обмежені. Перевагою керування за збуренням є те, що пристрій керування „не чекає” виникнення відхилення, а запобігає його появі. Остання обставина обумовлює більшу швидкодію й точність керування.

Найкращі результати досягаються при комбінованому керуванні. У цьому випадку для усунення впливу найбільш істотних збурень використовується керування за збуренням, а вплив усіх інших факторів на стан об'єкта керування контролюється за допомогою керування за відхиленням. Пристрій керування в цьому випадку буде мати більшу складність.

Все більш широке поширення отримує адаптивне керування об'єктами. При адаптивному керуванні здійснюється автоматична зміна параметрів, структури або алгоритму керування пристрою автоматичного керування для забезпечення обраного критерію оптимальності функціонування системи при доволі змінних зовнішніх впливах і змінних параметрах об'єкта. Адаптивне керування дозволяє або одержати стабільну якість керування, або оптимізувати керування в змінних умовах функціонування й при зміні характеристик об'єкта керування. Остання обставина досить актуальна при серійному випуску пристроїв керування, коли характеристики конкретного об'єкта керування заздалегідь невідомі.

Адаптивні системи керування за характером контрольованих змін можуть бути розділені на системи [35]:

- ▶ *із самоналаштуванням*, у яких є можливість здійснити тільки контрольовані зміни параметрів системи керування й керуючих впливів;
- ▶ *із самоорганізацією*, які дозволяють, крім того, виконувати контрольовані зміни структури системи керування;
- ▶ *із самоалгоритмізацією*, які, крім перерахованих вище змін, можуть виконувати й зміну алгоритму роботи;
- ▶ *із самонавчанням*, які, крім всіх вище перерахованих властивостей, здатні в результаті накопичення інформації (досвіду) виконувати зміну критерію за яким функціонує система.

Контрольовані зміни, які виконуються в системі під впливом зміни зовнішніх умов, можуть вибиратися з деякого набору функцій, записаних у програмі, або визначатися самою системою в результаті аналізу поточної інформації. Системи, у яких контрольовані зміни „запропоновані” на основі аналізу апріорної інформації в стадії проектування, називаються *пасивними*. Системи, де зміни заздалегідь не визначені, а „диктуються” тільки ситуацією яка склалася, називаються *активними*.

Оптимальний режим роботи в системі може розраховуватися на основі аналізу інформації. Такі системи називаються *аналітичними*. Якщо ж оптимальний режим визначається в результаті пошуку умов екстремуму, то системи називаються *пошуковими*.

Системи із самоналаштуванням можуть здійснювати зміну параметрів системи керування й керуючих впливів. Як правило, зміна параметрів системи керування приводить до зміни динамічних властивостей системи, а зміна керуючих впливів змінює лише координати робочої точки. Відповідно до цього системи із самоналаштуванням, що здатні здійснювати контрольовані зміни параметрів системи з метою одержання оптимальних у деякому сенсі

динамічних властивостей, називаються *динамічними системами із самоналаштуванням*.

2.3.4. Показники якості автоматичного регулювання

При автоматичному регулюванні мехатронної системи важливе значення мають показники якості регулювання, які оцінюють характер протікання перехідного процесу при регулюванні.

Швидкодія може характеризуватися часом регулювання t_p , за який вихідна змінна $y(t)$ величина перший раз досягне значення уставки $r=const$, часом першого максимуму t_{max} , коли відхилення вихідної змінної досягне першого максимуму, або загальним часом перехідного процесу $t_{n.n.}$ (рис. 2.9). На швидкодію, й взагалі на тривалість перехідного процесу $t_{n.n.}$, впливає також час запізнення початку руху t_c . Бажано, щоб швидкодія регулювання була якнайменшою (тобто $t_p \rightarrow \min$).

Коливальність характеризується часом затухання коливань вихідної змінної (це різниця $t_{n.n.} - t_p$). Цей показник повинен бути якнайменшим.

Перерегулювання характеризується максимальним відхиленням регульованої величини (вихідної змінної $y(t)$) від початкового її значення, віднесеному до заданого значення уставки $r=const$ регульованої величини. Цей параметр регулювання виражається у відносних одиницях:

$$\Delta y_{max}^* = \frac{\Delta y_{max}}{r_{const}}, \quad (2.4)$$

де r_{const} - значення уставки; Δy_{max} - максимальне відхилення регульованої величини від значення уставки r_{const} .

Показник перерегулювання повинен бути якнайменшим $\Delta y_{\max}^* \rightarrow \min$.

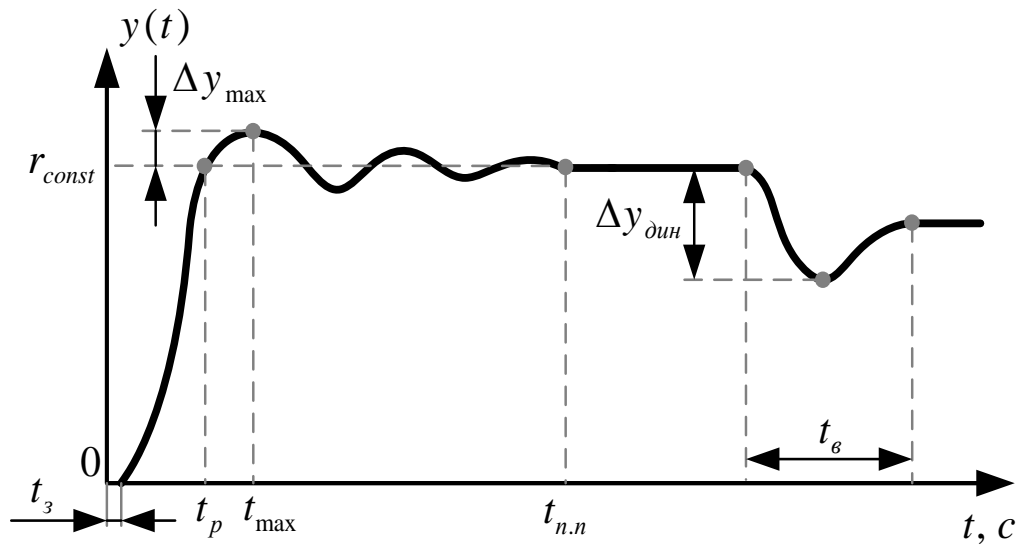


Рис. 2.9. Зміна вихідної величини $y(t)$ об'єкта керування при відпрацюванні уставки r_{const} та при дії на об'єкт керування зовнішнього збурення

Динамічна зміна регульованої величини $\Delta y_{\ddot{a}t}$ визначається максимальним відхиленням регульованої величини від її початкового значення в усталеному режимі. Необхідно, щоб $\Delta y_{\ddot{a}t} \rightarrow \min$.

Час відновлення вихідної величини t_ϵ характеризує тривалість процесу зміни вихідної величини при дії на об'єкт керування зовнішніх збурень. Бажано, щоб $t_\epsilon \rightarrow \min$ [36].

2.4. ПД-регулятор: теорія, особливості реальних пристроїв та методи їх налаштування

2.4.1. Історія виникнення ПД-регулятора

Серед усієї різноманітності структур регуляторів необхідно виділити окремий тип – пропорційно-інтегрально-диференціальний регулятор (ПД-

регулятор). В англійській термінології: proportional-integral-derivative controller (PID controller). Відомо: щонайменше 80 % регуляторів у промисловості – це ПІД-регулятори [37].

ПІД-регулятор призначений для вирішення завдань керування об'єктами з одним входом і одним виходом. Прикладом може бути сервопривод: завдання полягає в переміщенні механізму на задану відстань, якщо керуючим параметром є швидкість обертання двигуна.

Для розуміння ПІД-регулятора корисно звернутися до історії його виникнення. Регулятори цього типу відомі з кінця 19 століття. ПІД-регулятор у сучасному розумінні був запропонований і проаналізований у статті Миколи Мінорського (Nicolas Minorsky) в 1922 році. Командування флоту США найняло інженера російського походження Миколу Мінорського для розробки автоматичної системи керування рухом судна (ship steering system), яка б допомагала кермовому корабля й автоматично виконувала деякі його дії.

Завдання кермового полягало в тому, щоб вести корабель за прокладеним курсом – лінією на географічній карті моря. У результаті спостережень, було помічено, що кермовий, намагаючись повернути корабель на заданий курс, повертає штурвал судна тим сильніше, чим більше відхилення корабля від курсу. Формалізація цієї дії привела до наступних міркувань. Нехай e – відстань від корабля до лінії курсу: $e > 0$, якщо корабель відхилився від курсу вправо й $e < 0$, якщо корабель відхилився від курсу вліво, нехай u – поворот керма, причому якщо $u > 0$, то судно повертається ліворуч, якщо $u < 0$ – то вправо (рис. 2.10). Мінорський припустив, що можна використовувати наступний керуючий вплив:

$$u(t) = k \cdot e(t), \quad (2.5)$$

де $k > 0$ – деяка постійна величина.

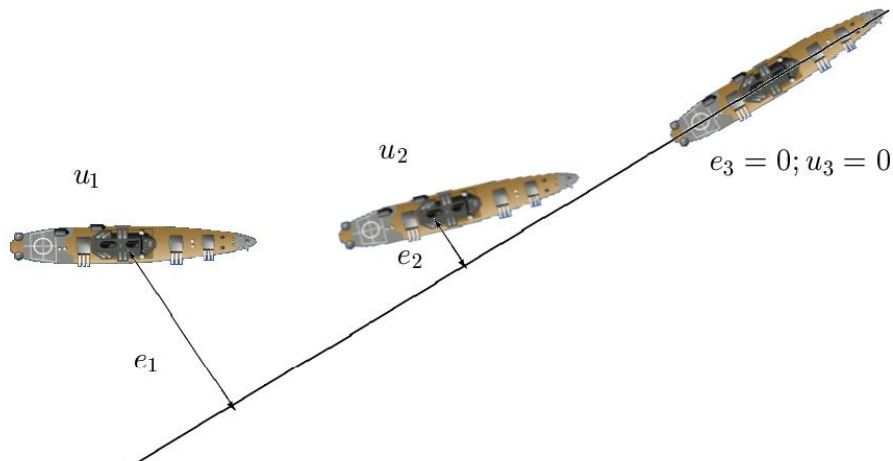


Рис. 2.10. Вихід корабля на курс при керуванні $u(t) = k \cdot e(t)$ (чим більше відхилення від курсу e , тим більший кут повороту керма u ($|e_1| > |e_2|$, $|u_1| > |u_2|$))

Неперервне керування кермом за таким законом приведе до того, що траєкторія руху судна буде нагадувати представлену на рис. 2.10. Чим ближче корабель до курсу, тем менше повернуто кермо й корабель плавно виходить на курс де $e=0$ і штурвал повертати вже нікуди не потрібно ($u=0$). Оскільки керуючий вплив постійно пов'язаний з відхиленням згідно виразу (2.5), то при „сході” корабля з курсу, відбудеться негайне повертання керма для корекції положення. Такий тип керування називається пропорційне керування – керуючий вплив u пропорційний похибці e .

Спостереження над роботою кермового були продовжені. Микола Мінорський установив, що досвідчений кермовий не тільки повертає штурвал судна виходячи з поточної похибки курсу, але й враховує також історію попередніх похибок положення корабля й швидкості зміни похибки курсу. Керування кораблем в умовах постійно присутніх збурень (наприклад, бічної течії) вимагає наступної техніки від кермового: якщо корабель довгий час іде паралельним курсом до прокладеного на карті, тобто похибка курсу не велика для ефективної дії пропорційного керування, але вона присутня постійно, то кермовий повертає штурвал убік лінії курсу тем сильніше, чим довше триває

похибка. Формалізація цієї дії приводить до керування, у якому керуючий вплив є сумою двох компонентів:

$$u(t) = k_D \cdot e(t) + k_I \cdot \int_0^t e(t) dt, \quad (2.6)$$

де $k_D > 0$ та $k_I > 0$ – деякі постійні, вибір яких визначає поведінку регулятора (пропорційний та інтегральний коефіцієнти ПІД-регулятора відповідно).

Доданок, що визначає в керуючому впливі інтеграл від похибки, називається інтегральним керуванням: сигнал похибки накопичується за весь час роботи системи й навіть невелика, але тривало діюча похибка, приведе до її впливу, який буде скомпенсований.

Інтегральне керування призначене для підвищення точності керування й дозволяє боротися з постійно діючими похибками. Відомо, що теоретично звести похибку регулювання до нуля у завданні установки виходу в задане значення можливо тільки при застосуванні інтегрального керування – пропорційне керування завжди залишає ненульову похибку. Проблема застосування інтегрального керування полягає в тому, що в процесі регулювання керуючий вплив помітно коливається біля шуканого постійного значення. Це пояснюється тим, що накопичена похибка у доданку $k_I \cdot \int_0^t e(t) dt$ із

виразу (2.6), перетворюється в нуль коли сигнал похибки змінить свій знак і тривалий час проінтегрується. Тому реалізується наступний сценарій: якщо якийсь час похибка була позитивною $e > 0$, то накопичена похибка приводить до позитивного керуючого впливу $u > 0$, але навіть коли похибка в результаті буде скомпенсована ($e = 0$), то інтегральна складова продовжить діяти й вихід системи перевищить бажаний (стане $e < 0$), що буде тривати поки від'ємний сигнал похибки в при інтегруванні не компенсує накопичену до цього позитивну

похибку. У результаті виникають тривалі загасаючі коливання керуючої змінної, що небажано з погляду якості керування.

Для вирішення цієї проблеми в керуючий вплив додають ще один компонент, який „пригнічує” інтегральне керування при значних коливаннях сигналу похибки, залишаючи його дію тільки при малих і тривалих похибках. Цей компонент – диференціальне керування, яке входить як третій доданок:

$$u(t) = k_D \cdot e(t) + k_I \cdot \int_0^t e(t)dt + k_D \cdot \frac{d}{dt} e(t), \quad (2.7)$$

де k_D – диференціальний коефіцієнт ПД-регулятора.

При роботі ПД-регулятора, звичайно основну частину керуючого впливу становить пропорційний компонент. Інтегральний і диференціальний компоненти призначені, в основному, для поліпшення точності і якості перехідного процесу – вони відіграють допоміжну роль. Значення коефіцієнтів суттєво залежать від поведінки об’єкта керування.

2.4.2. Класичний ПД-регулятор та його частинні випадки

2.4.2.1. Загальна структура системи із регулятором

ПД-регулятори широко застосовуються в мехатронних системах. Вони є невід’ємною частиною таких пристроїв як частотні перетворювачі для керування двигунами, температурні контролери (термостати), системи кондиціонування тощо. ПД-регулятор може бути реалізований як окремий прилад для установки в шафу автоматики, що має стандартні промислові інтерфейси 4-20 мА й 0-10 В [34].

Окрім, власне, ПД-регулятора в системі керування повинен бути формувач сигналу похибки:

$$\dot{a}(t) = r(t) - y(t), \quad (2.8)$$

де $r(t)$ - вхідний сигнал об'єкта керування (завдання, уставка); $y(t)$ - вихідний сигнал об'єкта керування (поточне або реально виміряне значення величини, яка регулюється). Звідси маємо структуру системи із зворотнім зв'язком (замкнена система керування), яка показана на рис. 2.11.



Рис. 2.11. Структура системи із застосуванням ПІД-регулятора

2.4.2.2. Закони регулювання

Під законом регулювання розуміють математичний вираз, що встановлює зв'язок формованого регулятором впливу (керування u) на об'єкт зі змінними, інформація про які при цьому використовується. У якості таких змінних звичайно виступають, регульована величина, зовнішні впливи (збурення) та уставка (завдання).

Багаторічна практика автоматизації показала, що переважна більшість завдань може бути вирішена за допомогою регуляторів по відхиленню, що враховують поточне значення відхилення (похибки), першу похідну похибки й інтеграл від похибки за час регулювання. З'явилися поняття про стандартні закони регулювання у вигляді математичних зв'язків між регулюючим впливом і похибкою регулювання.

Простота налаштування стандартних регуляторів і достатньо висока якість регулювання забезпечили їм широке поширення в промисловості та сільському господарстві. Формований закон регулювання, а також показники точності його

формування стали основними характеристиками в паспорті стандартного промислового регулятора. Отже, найважливіше міркування при виборі того або іншого конкретного регулятора - формований ним закон регулювання. Стандартні закони регулювання поділяються на лінійні й нелінійні. Розглянемо ці закони, приділивши увагу рекомендаціям з їх застосування.

Найбільш складним стандартним лінійним законом регулювання є ПД-закон (3). Коефіцієнти k_D , k_D та k_I , які враховують питому вагу пропорційної, диференціальної та інтегральної складової відповідно у формованому сигналі регулювання, є параметрами закону регулювання і можуть змінюватися в широких межах відповідними налаштуваннями промислового регулятора.

Інші стандартні лінійні закони регулювання є різними частинними випадками ПД - закону.

П-закон.

Найпростішим лінійним стандартним законом регулювання є пропорційний або П-закон (2.5). Очевидно, це окремий випадок ПД-закону при $k_D = 0$ та $k_I = 0$. Можливості цього регулятора у формуванні характеристик і властивостей автоматичної системи мінімальні. У більшості випадків, за винятком випадків астатичних об'єктів, його застосування веде до побудови статичних систем автоматичного регулювання, які здатні працювати тільки в позиційному (статичному) режимі, причому з деякою похибкою статизму. Враховуючи наведені вище властивості, П-регулятор може бути рекомендований для побудови найпростіших систем регулювання мехатронних систем, тобто у випадках простих об'єктів регулювання й режимів роботи системи, а також невисоких вимог до якості (точності, швидкодії, запасу стійкості).

I-закон.

Інший стандартний закон регулювання, який може бути також віднесений до найпростіших, це інтегральний або I-закон :

$$u(t) = k_I \cdot \int_0^t e(t) dt = \frac{1}{T_I} \cdot \int_0^t e(t) dt, \quad (2.9)$$

де T_I - постійна часу інтегрування. Він являє собою окремий випадок закону (2.7) для $k_D = 0$ та $k_P = 0$. Застосування цього регулятора дозволяє будувати астатичні системи регулювання, придатні для роботи не тільки в позиційному, але й у кінетичному режимі похибки. Однак, небажаною властивістю цього закону є знижена швидкодія. Інтегральні регулятори застосовуються також у простих системах, у випадках, коли переваги астатичної системи значать більше, ніж виникаючі ускладнення, а це надзвичайно рідкі випадки.

ПІ-закон.

Випадок $k_D = 0$ у виразі (2.7) приводить до пропорційно-інтегрального або ПІ-закону регулювання. Його параметрами є коефіцієнт пропорційності k_P і постійна часу інтегрування T_I .

ПІ-регулятори одержали винятково широке поширення завдяки відносній простоті й вдалій комбінації позитивних властивостей інтегральних і пропорційних регуляторів. Вони дозволяють будувати астатичні системи регулювання, але не ускладнюють проблеми стабілізації системи. У результаті вдається одержати гарну точність і швидкодію.

ПД-закон.

Якщо у виразі (2.7) прийняти $k_I = 0$, то ми отримаємо пропорційно-диференціальний, або ПД-закон регулювання:

$$u(t) = k_D \cdot e(t) + \dot{O}_D \cdot \frac{d}{dt} e(t) = k_D \cdot e(t) + \dot{O}_D \cdot \frac{d}{dt} e(t), \quad (2.10)$$

де \dot{O}_D - постійна часу диференціювання. Його застосування веде до побудови статичних систем регулювання з негативними наслідками, що впливають із

цього факту. Найчастіше ПД-регулятор застосовується для підвищення швидкодії.

ПД-закон.

Найзагальнішим, як це було вказано вище, є ПД-регулятор. В число його параметрів входять коефіцієнт пропорційності k_D , постійна часу інтегрування T_I та постійна часу диференціювання \dot{D} .

Даний регулятор дозволяє будувати астатичні системи автоматичного регулювання. У порівнянні з П-регулятором він дає ще додаткове форсування (збільшення швидкодії) й із усіх стандартних регуляторів надає найбільші можливості формування бажаних характеристик і властивостей системи. Його застосовують у найбільш відповідальних і складних випадках, тобто коли об'єкт описується диференціальним рівнянням високого порядку й має значну інерційність, або коли ставляться високі вимоги до якості руху мехатронної системи. На практиці зустрічається чимало завдань регулювання руху мехатронних систем, коли навіть можливості ПД-регулятора виявляються недостатніми. Проте, і вони можуть бути вирішені на базі регуляторів, що формують стандартні закони регулювання. При цьому використовують наступні прийоми:

- ▶ каскадне включення регуляторів;
- ▶ багатоконтурна побудова системи з використанням належних локальних регуляторів у кожному контурі;
- ▶ автоматична зміна параметрів, тобто побудова адаптивних систем [38].

2.4.3. Особливості реальних регуляторів

Описаний вище ПД-регулятор є теоретичною ідеалізацією реальних регуляторів, тому для їхнього практичного використання необхідно врахувати

особливості реальних умов їх застосування і технічної їх реалізації. До таких особливостей відносяться:

- ▶ обмежений динамічний діапазон змін фізичних змінних у системі (наприклад, обмежена потужність двигуна, обмежена пропускна здатність клапана);
- ▶ обмежена точність вимірів, що вимагає спеціальних заходів для виконання операцій диференціювання з прийнятною похибкою;
- ▶ наявність практично у всіх системах типових нелінійностей: насичення (обмеження динамічного діапазону зміни змінних), обмеження швидкості наростання, гістерезис, люфт;
- ▶ технологічна і випадкова варіація параметрів регулятора та об'єкта керування;
- ▶ дискретна реалізація регулятора;
- ▶ необхідність плавного перемикавання режимів регулювання.

Опишемо декілька методів вирішення вказаних проблем.

Проблема чисельного диференціювання та шуми.

Проблема чисельного диференціювання є досить старою й загальною як у цифрових, так і в аналогових регуляторах. Суть її полягає в тому, що похідна обчислюється звичайно як різниця двох близьких по величині значень функції, тому відносна похибка похідної завжди виявляється більшою, ніж відносна похибка чисельного представлення функції, яка диференціюється.

Зокрема, якщо на вхід диференціатора надходить синусоїдальний сигнал $A \sin(\omega t)$, то на виході одержимо $A \omega \cos(\omega t)$, тобто з ростом частоти ω збільшується амплітуда сигналу на виході диференціатора. Інакше кажучи, диференціатор підсилює високочастотні перешкоди, короткі викиди й шум. Крім шумів диференціювання, на характеристики ПІД-регулятора впливають

шуми вимірів. Високочастотні шуми шкідливі тим, що викликають швидке зношування електродвигунів.

Оскільки об'єкт керування звичайно є низькочастотним фільтром, шуми вимірів рідко проходять по контуру регулювання на вихід системи. Однак вони збільшують похибку вимірів і знижують точність регулювання. У ПД-регуляторах розрізняють шум з низькочастотним спектром, викликаний зовнішніми впливами на об'єкт керування, і високочастотний шум, пов'язаний з електромагнітними перешкодами у кабелях живлення, з дискретизацією вимірюваного сигналу й іншими причинами. Низькочастотний шум моделюють як зовнішні збурення, високочастотний - як шуми вимірів.

Звичайно для зниження шкідливої дії високочастотних сигналів використовують фільтри високих частот, які не пропускають на вхід ПД-регулятора сигнали високих частот.

Інтегральне насичення

В усталеному режимі роботи й при малих збуреннях більшість систем з ПД-регуляторами є лінійними. Однак процес виходу на усталений режим практично завжди вимагає врахування нелінійності типу „обмеження”. Ця нелінійність пов'язана із природними обмеженнями на потужність, швидкість, частоту обертання, кут повороту, площу поперечного перерізу клапана, динамічний діапазон тощо. Контур регулювання в системі, що перебуває в насиченні (коли змінна досягла обмеження), виявляється розімкнутим, оскільки при зміні змінної на вході ланки з обмеженням його вихідна змінна залишається без змін.

Найбільш типовим проявом режиму обмеження є так зване інтегральне насичення, яке виникає в процесі виходу системи на усталений режим у регуляторах з ненульовим коефіцієнтом k_I . Суть проблеми інтегрального насичення полягає в тому, що якщо сигнал на вході об'єкта керування $u(t)$

увійшов у зону насичення (обмеження), а сигнал похибки $r(t)-y(t)$ не дорівнює нулю, інтегратор продовжує інтегрувати, тобто сигнал на його виході росте, але цей сигнал не бере участі у процесі регулювання й не впливає на об'єкт внаслідок ефекту насичення. Система керування в цьому випадку стає еквівалентною розімкнутій системі, сигнал на вході якої дорівнює рівню насичення керуючого сигналу.

Ефект інтегрального насичення відомий давно. В аналогових регуляторах його усунення було досить проблематичним, оскільки в них проблема не могла бути вирішена алгоритмічно, а тільки апаратними засобами. З появою мікропроцесорів проблему вдається розв'язати набагато ефективніше. Методи усунення інтегрального насичення звичайно є предметом винаходів, вони відносяться до комерційної таємниці фірм-виробників і захищаються патентами. Нижче розглянуто кілька таких ідей, описаних у літературі:

1. **Обмеження швидкості наростання вхідного впливу.** Оскільки максимальне значення вхідного впливу на об'єкт керування $u(t)$ знижується зі зменшенням різниці $r(t)-y(t)$, то для усунення ефекту обмеження можна просто знизити швидкість наростання сигналу уставки $r(t)$, наприклад, за допомогою фільтра. Недоліком такого способу є зниження швидкодії системи, а також неможливість усунути інтегральне насичення, викликане зовнішніми збуреннями, а не сигналом уставки;
2. **Алгоритмічна заборона інтегрування.** Коли керуючий вплив на об'єкт досягає насичення, зворотний зв'язок розривається, а інтегральна складова продовжує рости. Тому один з методів усунення інтегрального насичення полягає в тому, що контролер стежить за керуючим впливом на об'єкт, і як тільки він досягає насичення, контролер вводить програмну заборону інтегрування для інтегральної складової.

Безударне перемикання режимів регулювання.

У ПІД-регуляторах можуть існувати режими, коли їх параметри змінюються стрибком. Наприклад, коли в працюючій системі треба змінити коефіцієнт k_I або якщо після ручного керування системою необхідно перейти на автоматичний режим. В описаних випадках можуть з'явитися небажані викиди (стрибки) регульованої величини, якщо не прийняти спеціальних заходів. Тому виникає завдання плавного (безударного) перемикання режимів роботи або параметрів регулятора. Основний метод вирішення проблеми полягає в побудові такої структури регулятора, коли зміна параметра виконуються до етапу інтегрування. Наприклад, при змінному параметрі $k_I(t)$ інтегральний член можна записати у двох формах:

$$I(t) = k_I(t) \int_0^t e(t) dt \quad \text{або} \quad I(t) = \int_0^t k_I(t) e(t) dt. \quad (2.11)$$

У першому випадку при стрибкоподібній зміні $k_I(t)$ інтегральний член буде мінятися стрибком, у другому випадку - плавно, оскільки $k_I(t)$ перебуває під знаком інтеграла, значення якого не може змінюватися стрибком.

Дискретна форма регулятора.

Неперервні змінні зручно використовувати для аналізу й синтезу ПІД-регуляторів. Для технічного втілення необхідно перейти до дискретної форми рівнянь, оскільки основою всіх регуляторів є мікроконтролер, контролер або комп'ютер, які оперують зі змінними, отриманими з аналогових сигналів після їхньої дискретизації за часом і квантуванням за рівнем. Внаслідок деякої тривалості часу обчислення керуючого впливу в мікроконтролері й затримки аналого-цифрового перетворення між моментом надходження аналогового сигналу на вхід регулятора й появою керуючого впливу на його виході з'являється небажана затримка, яка збільшує загальну затримку в контурі регулювання й знижує запас стійкості. Одним із варіантів вирішення проблем, які виникають при перетворенні аналогових сигналів у дискретну форму у ПІД-

регуляторах є збільшення частоти дискретизації так, щоб вона принаймні в 2 рази була вищою, ніж максимальна частота перешкоди. Це дозволяє застосувати після дискретизації цифровий фільтр нижніх частот. При такій частоті дискретизації отриманий цифровий сигнал з погляду кількості інформації повністю еквівалентний аналоговому й усі властивості аналогового регулятора можна поширити на цифровий [39].

2.4.4. Методи налаштування ПІД-регуляторів

2.4.4.1. Ручні методи

Найпростішим способом визначення параметрів ПІД-регулятора може бути наступна стратегія. Спочатку необхідно обнулити усі його коефіцієнти ($k_P=0$, $k_D=0$, $k_I=0$), що рівнозначно відключенню ПІД-регулятора від об'єкта керування. Після чого починають працювати з типовим впливом – сходиною: на вхід $r(t)$ у якийсь момент часу подається постійний сигнал заданої амплітуди $r(t) = const$. Задача налаштування регулятора полягає у тому, щоб вихідна величина $y(t)$ об'єкта керування якнайкраще повторювала вхідний вплив $r(t)$. Для цього поступово збільшують k_P , постійно перевіряючи реакцію на вплив $r(t) = const$, доти, поки вихідна величина $y(t)$ не почне коливатися – це критичне значення коефіцієнта k_P^{\max} . Після цього необхідно встановити остаточне значення коефіцієнта $k_P = \frac{k_P^{\max}}{2}$. Якість регулювання можна покращити, налаштуваючи коефіцієнти k_D та k_I : якщо залишкова похибка занадто велика, то необхідно незначно збільшити k_I , якщо сильне перерегулювання – збільшити k_D .

Крім налаштування ПІД-регулятора можна виконати його підлаштування, яке засноване на певних правилах. Ці правила отримані з досвіду, теоретичного аналізу й чисельних експериментів. Вони зводяться до наступного:

- ▶ збільшення пропорційного коефіцієнта k_p збільшує швидкодію й знижує запас стійкості;
- ▶ зі зменшенням пропорційного коефіцієнта k_p похибка регулювання із часом зменшується швидше;
- ▶ збільшення коефіцієнта k_I зменшує запас стійкості;
- ▶ збільшення k_D збільшує запас стійкості й швидкодію.

Перераховані правила застосовуються також для регуляторів, що використовують методи експертних систем і нечіткої логіки.

Ручне налаштування за допомогою правил зручно виконувати із застосуванням інтерактивного програмного забезпечення на комп'ютері, тимчасово включеному в контур керування. Для оцінки реакції системи на зміну уставки, зовнішні впливи або шуми вимірів подають штучні впливи, які їх імітують, й спостерігають реакцію на них. Після виконання налаштування значення коефіцієнтів регулятора записують у пам'ять ПД-контролера, а комп'ютер відключають.

Відзначимо, що застосування приведених вище правил можливо тільки після попереднього налаштування регулятора за формулами. Спроби налаштувати регулятор без початкового наближення, виконаного за розрахунками відповідних коефіцієнтів регулятора, можуть бути безуспішними. Сформульовані вище правила справедливі тільки в околі оптимального налаштування регулятора.

2.4.4.2. Метод Зіглера-Нікольса

Вперше методику розрахунків параметрів ПД-регулятора запропонували Зіглер і Нікольс в 1942 році. Ця методика дуже проста й не дає оптимальних результатів, у загальному випадку. Проте, вона дотепер часто використовується на практиці, хоча з тих пір з'явилась велика кількість більш точних методів. Основний принцип визначення коефіцієнтів ПД-регулятора полягає в тому,

щоб керованому об'єкту дати типовий керуючий вплив (наприклад, подати постійний сигнал $r(t)=1$) і подивитися на його реакцію. Тобто цей метод відноситься до емпіричних і заснований на використанні даних, отриманих експериментально на реальному об'єкті. Одержали поширення два варіанти методу.

Перший варіант заснований на використанні запасу стійкості. У цьому варіанті використовуються результати експериментального дослідження системи, що складається із П-регулятора й заданого об'єкта регулювання. Коефіцієнт k_P П-регулятора збільшується доти, поки на виході системи не встановляться коливання з постійною амплітудою коливань, тобто система не виявиться на межі стійкості. Фіксується й позначається через k_P^* значення коефіцієнта передачі регулятора, при якому система перебуває на межі стійкості. Вимірюється тривалість T^* періоду коливань, які встановилися в системі. Значення параметрів регулятора обраного типу розраховуються за формулами, наведеними у табл. 2.1. Недоліком методу є необхідність виводити систему на межу стійкості, що для багатьох об'єктів не рекомендується.

Таблиця 2.1. Налаштування коефіцієнтів різних регуляторів для першого варіанту методу Зіглера-Нікольса

	k_P	k_I	k_D
П-регулятор	$0,5k_P^*$	-	-
ПІ-регулятор	$0,5k_P^*$	$0,54k_P^* / \dot{\sigma}^*$	-
ПІД-регулятор	$0,6k_P^*$	$1,2k_P^* / \dot{\sigma}^*$	$0,075 k_P^* \dot{\sigma}^*$

У **другому варіанті** використовується реакція об'єкта керування на східчастий вплив ($r=1$), яка називається кривою розгону (рис. 2.12). Якщо крива

розгону має аперіодичний характер, то об'єкт керування має таку передаточну функцію:

$$W(s) = \frac{k_0}{T_0 s + 1} e^{-\tau s}, \quad (2.12)$$

де k_0 - коефіцієнт передачі; T_0 - постійна часу; τ - час запізнення.

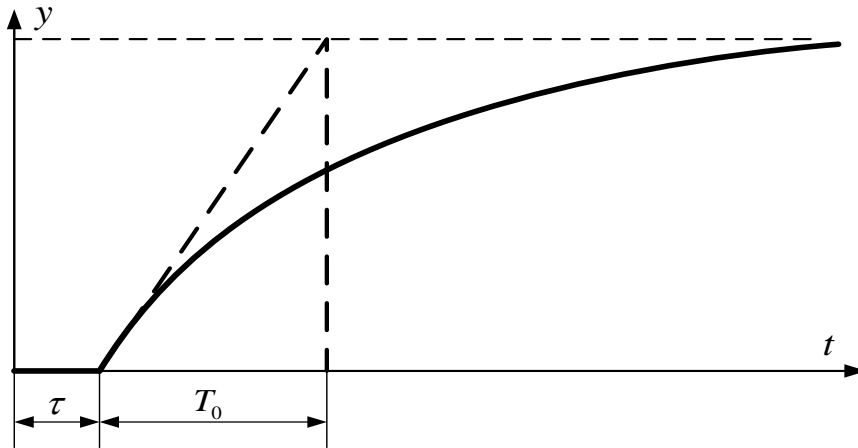


Рис. 2.12. Крива розгону об'єкта керування

Параметри регулятора розраховуються за формулами, наведеними у табл.

2.2. Метод дає гарні результати, якщо $0,15 < \tau T < 0,6$.

Таблиця 2.2. Налаштування коефіцієнтів різних регуляторів для другого варіанту методу Зіглера-Нікольса

	k_P	k_I	k_D
П-регулятор	$\dot{O}_0 / k_0 \tau$	-	-
ПІ-регулятор	$0,9 \dot{O}_0 / k_0 \tau$	$0,3 \dot{O}_0 / k_0 \tau^2$	-
ПІД-регулятор	$1,2 \dot{O}_0 / k_0 \tau$	$0,6 \dot{O}_0 / k_0 \tau^2$	$0,6 \dot{O}_0 / k_0$

Регулятори, параметри яких розраховані по методу Зіглера-Нікольса, не завжди забезпечують необхідну якість процесу регулювання. Як правило,

потрібне додаткове налаштування їх параметрів. Незважаючи на це, метод Зіглера-Нікольса й деякі його модифікації досить популярні, і багато виробників регуляторів рекомендують їх для налаштування регуляторів [40].

2.4.4.3. Автоматичне налаштування та адаптація

Природнім напрямком розвитку комерційних ПД-регуляторів є розробка методів, що дозволяють знизити витрати людської праці на їхню інсталяцію, налаштування й обслуговування. Незважаючи на те, що багато методів автоматичного налаштування й адаптації ПД-регуляторів, що використовуються в цей час, були розроблені ще в 60-х роках, у промислових контролерах адаптивна техніка почала використовуватися тільки із середини 80-х. Це пов'язано з технічною складністю реалізації адаптивних алгоритмів на елементній базі, яка існувала до появи мікроконтролерів.

Налаштування може виконуватися **вручну або автоматично**, без участі людини (**автоналаштування**).

Автоналаштування може виконуватися повністю автоматично й на вимогу, коли людина є ініціатором налаштування. Повністю автоматичне налаштування може ініціюватися при настанні заздалегідь заданої умови, наприклад, при зміні навантаження, при зміні зовнішніх впливів, при зміні похибки регулювання або безперервно в часі. Автоматичне налаштування, яке ініціюється без участі людини, називається адаптацією. Іноді термін „адаптація” трактують більш широко як пристосування регулятора до реального об'єкта на стадії введення системи в експлуатацію.

Різновидом адаптації є розімкнуте керування параметрами регулятора (табличне налаштування), коли заздалегідь знайдені параметри регулятора для різних умов роботи системи заносяться в таблицю, з якої вони зчитуються при настанні умов, за якими ініціюється адаптація.

Відзначимо, що регулятори, налаштовані в автоматичному режимі, частіше налаштовані гірше, ніж в ручному режимі. Пояснюється це філософським міркуванням, що комп'ютер не може виконувати складні й погано формалізовані завдання краще людини. У даний момент відсутні прості, надійні й загальноприйняті методи автоматичного налаштування.

Усі види автоматичного налаштування використовують три принципово важливі етапи: ідентифікація, розрахунки параметрів регулятора, налаштування. Часто кінцевий етап включає етап підлаштування (оптимізація налаштування). Оптимізацію налаштування необхідно виконувати у зв'язку з тим, що методи розрахунків параметрів регулятора за формулами є спрощеними, не враховують нелінійності об'єкта, (наприклад, завжди присутню у мехатронних системах нелінійність типу „обмеження”), а ідентифікація параметрів об'єкта виконується з деякою похибкою. Підлаштування регулятора може бути пошуковим (без ідентифікації об'єкта, шляхом пошуку оптимальних параметрів) і безпошуковим (з ідентифікацією). Пошукове підлаштування базується звичайно на правилах або на ітераційних алгоритмах пошуку мінімуму критеріальної функції. Найпоширеніший пошук оптимальних параметрів - за допомогою градієнтного методу пошуку. Для цього знаходять похідні від критеріальної функції за параметрами ПІД-регулятора, які є компонентами вектора градієнта. Далі проводиться зміна параметрів відповідно до напрямку градієнта.

Важливо підкреслити, що незважаючи на наявність автоматичного налаштування, контролер може не дати необхідної якості регулювання із причин, що не залежать від якості закладених у нього алгоритмів. Наприклад, об'єкт керування може бути погано спроектований (залежні контури регулювання, велика затримка, високий порядок об'єкта); об'єкт може бути нелінійним; датчики можуть бути розташовані не в тому місці, де потрібно, і мати поганий контакт із об'єктом; рівень перешкод у каналі виміру може бути

неприпустимо великим; точність датчика може бути недостатньо високою; джерело вхідного впливу на об'єкт може мати занадто велику інерційність або гістерезис; можуть бути також помилки у монтажі системи, погане заземлення, обриви провідників тощо.

Тому перш ніж починати автоматичне налаштування, необхідно переконатися у відсутності перерахованих проблем. Наприклад, внаслідок зношування механічної системи з'явився непередбачуваний люфт і тому система перебуває в режимі коливань: підлаштування регулятора може не дати бажаного результату, поки не усунута причина проблеми.

Структурна схема системи, яка самоналаштовується наведена на рис. 2.13. Автоналаштування практично не має ніяких особливостей у порівнянні з описаними раніше методами, за винятком того, що воно виконується в автоматичному режимі.

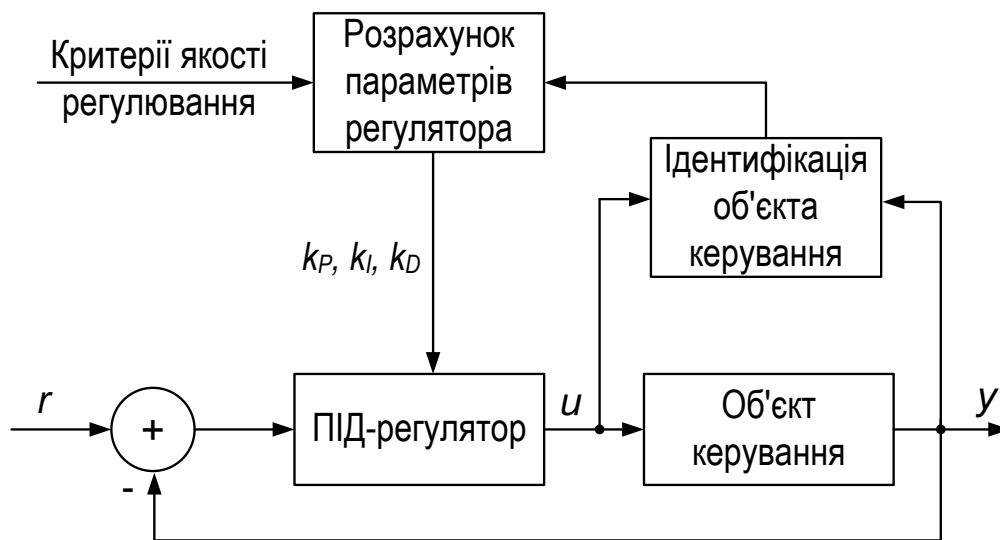


Рис. 2.13. Загальна структура системи із автоматичним налаштуванням

Основним етапом автоматичного налаштування й адаптації з без пошуковим підлаштуванням є ідентифікація моделі об'єкта. Вона виконується в автоматичному режимі звичайними методами.

Більшість промислових ПІД-регуляторів містять функції автоматичного налаштування, що дозволяють автоматично визначити оптимальні параметри k_P , k_D , k_I . Автоматичне налаштування може бути повторено, якщо якість керування не відповідає бажаній [40].

2.4.4.4. Табличне налаштування регулятора

Найбільш простим методом адаптації ПІД-регулятора до змінних властивостей об'єкта керування є метод табличного керування коефіцієнтами регулятора. Він може бути використаний не тільки для адаптивного керування, але й для керування нелінійними об'єктами, нестационарними процесами або при необхідності змінювати параметри залежно від деяких умов. Принцип табличного керування дуже простий (рис. 2.14).

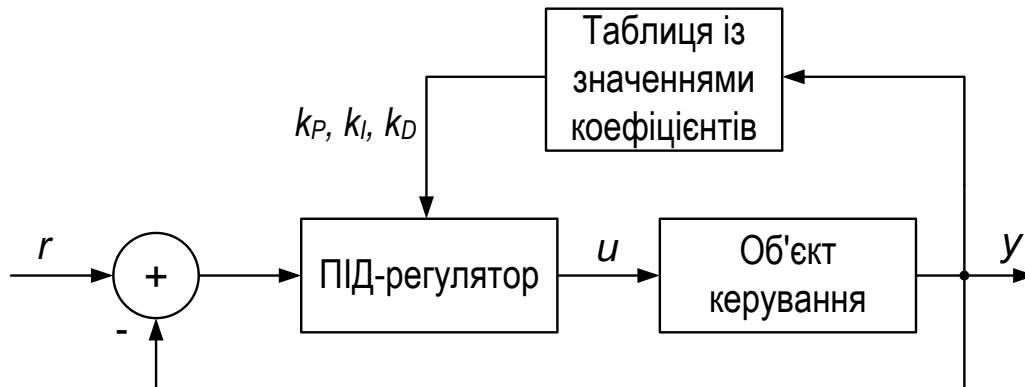


Рис. 2.14. Адаптивне керування за допомогою параметрів занесених у таблицю

Знаючи заздалегідь можливі зміни режиму роботи системи (наприклад, можливі варіанти навантаження електродвигуна мехатронної системи), виконують ідентифікацію об'єкта для декількох різних режимів і для кожного з них знаходять параметри регулятора. Значення цих параметрів записують у таблицю. У процесі функціонування системи вимірюють параметри, які характеризує режим роботи системи (наприклад, струм, що споживається

електродвигуном), і залежно від їхніх значень вибирають з таблиці коефіцієнти ПД-регулятора.

Описану систему можна розглядати як систему із двома контурами регулювання. Однак контур, що служить для адаптації, у нашому прикладі з струмом, який споживається електродвигуном, є розімкнутим. Із цієї причини табличне керування характеризується високою швидкістю, відсутністю неправильного спрацьовування або розбіжності алгоритмів адаптації.

У загальному випадку таблиця може бути багатомірною. Для зниження необхідної розмірності таблиці можна використовувати інтерполяцію даних за допомогою сплайна.

Недоліком цього методу є більші витрати на попередню ідентифікацію, необхідну для складання таблиці. Однак цю проблему частково можна розв'язати застосуванням спеціальних програмних засобів для автоматичної ідентифікації й адаптації регулятора, після виконання яких отримані параметри заносяться в таблицю.

Табличне керування коефіцієнтами регулятора доцільно використовувати в тих випадках, коли заздалегідь відомі види й значення дестабілізуючих факторів, які до того ж можна виміряти [40].

2.5. Оптимальне керування мехатронними системами

2.5.1. Вступ. Історичний розвиток екстремальних задач

Задачі оптимального керування відносяться до теорії екстремальних задач, тобто задач визначення максимальних і мінімальних значень. Уже та обставина, що в цьому реченні зустрічається кілька латинських слів (maximum - найбільше, minimum - найменше, extremum - крайнє, optimus - оптимальне), вказує, що теорія екстремальних задач була предметом дослідження із прадавніх часів. Про деякі з таких задач писали ще Аристотель (384-322 роки до

н.е.), Евклід (III в. до н.е.) і Архімед (287-212 роки до н.е.). Заснування міста Карфаген (825 рік до н.е.) легенда асоціює з найдавнішою задачею визначення замкненої плоскої кривої, що охоплює фігуру максимально можливої площі. Подібні задачі іменуються ізопериметричними. Характерною рисою екстремальних задач є те, що їх постановка була породжена актуальними запитами розвитку суспільства. Більше того, починаючи з XVII століття домінуючим твердженням стає те, що закони навколишнього світу є наслідком деяких варіаційних принципів. Першим з них був принцип П. Ферма (1660 рік), відповідно до якого траєкторія світла, що поширюється від однієї точки до іншої, повинна бути така, щоб час проходження світла уздовж цієї траєкторії був мінімально можливим. Згодом були запропоновані різні широко використовувані в природознавстві варіаційні принципи, наприклад: принцип стаціонарної дії Гамільтона (1834 рік), принцип віртуальних переміщень, принцип найменшого примусу тощо. Паралельно розвивалися й методи розв'язування екстремальних задач. Близько 1630 року Ферма сформулював метод дослідження на екстремум поліномів, який полягає в тому, що в точці екстремуму похідна функції дорівнює нулю. Для загального випадку цей метод був отриманий Ньютоном (1671) і Лейбніцем (1684), роботи яких знаменують зародження математичного аналізу.

Початок розвитку класичного варіаційного числення датується появою в 1696 році статті І. Бернуллі (учня Лейбніця), у якій сформульована постановка задачі про криву, що з'єднує дві точки А і В, рухаючись по якій із точки А в В під дією сили ваги матеріальна точка досягне В за мінімально можливий час. У рамках класичного варіаційного числення в XVIII-XIX століттях установлені необхідні умови екстремуму першого порядку (Ейлер, Лагранж), пізніше досліджені необхідні й достатні умови другого порядку (Вейерштрасс, Лежандр, Якобі), побудовані теорія Гамільтона-Якобі й теорія поля (Д. Гільберт, А. Кнезер).

Подальший розвиток теорії екстремальних задач привів в ХХ столітті до створення лінійного програмування, випуклого аналізу, математичного програмування, теорії мінімаксу й деяких інших розділів, одним з яких є теорія оптимального керування. Ця теорія подібно іншим напрямкам теорії екстремальних задач, виникла у зв'язку з актуальними задачами автоматичного регулювання наприкінці 40-х років (керування ліфтом у шахті з метою найшвидшої його зупинки, керування рухом ракет, стабілізація потужності гідроелектростанцій тощо).

Відмітимо, що постановки окремих задач, які можуть бути інтерпретовані як задачі оптимального керування, зустрічалися й раніше, наприклад в „Математичних початках натуральної філософії” І. Ньютона (1687). Сюди ж відносяться й задачі Р. Годдарда (1919) про підйом ракети на задану висоту з мінімальними витратами палива й обернена їй задача про підйом ракети на максимальну висоту при заданій кількості палива [41].

За минулий час були встановлені фундаментальні принципи теорії оптимального керування: принцип максимуму [42] й метод динамічного програмування [43]. Зазначені принципи являють собою розвиток класичного варіаційного числення [44] для дослідження задач, що містять складні обмеження на керування. Зараз теорія оптимального керування переживає період бурхливого розвитку як у зв'язку з наявністю цікавих математичних проблем, так і у зв'язку з наявністю сфер її застосування, у тому числі й у таких областях як економіка, біологія, медицина, ядерна енергетика тощо.

2.5.2. Постановка задач оптимального керування та їх класифікація

Оптимальним називають найкраще в деякому сенсі керування. У більшості випадків перевести об'єкт керування з одного стану в інший (з вихідного в заданий) можна безліччю способів. Ці способи реалізуються за допомогою різних законів керування. Часто серед них можна вибрати такий

закон, щоб перехідний процес був оптимальним за певним критерієм (критерієм оптимальності). У якості критерію може виступати, наприклад, мінімум енергії, що витрачається на процес переходу або мінімум часу переходу. Критерій оптимальності формалізується у вигляді деякого функціонала, екстремум якого (мінімум або максимум) свідчить, що перехідний процес і керування оптимальні.

Загальний вид функціонала такий:

$$J = g_0(X(t_0), X(t_\varepsilon), t_0, t_\varepsilon) + \int_{t_0}^{t_\varepsilon} f_0(X(t), U(t), t) dt, \quad (2.13)$$

де X – вектор змінних стану об’єкта керування; U – вектор керуючих впливів; t_0 , t_ε – початковий і кінцевий моменти часу перехідного процесу.

Функція g_0 визначає „якість” крайових станів, у тому числі, пов’язаних з величинами t_0 та t_ε . Функція f_0 визначає „якість” траєкторій $X(t)$ і керування $U(t)$ на інтервалі $[t_0; t_\varepsilon]$.

Задача, у якій відшукується екстремум функціонала (2.13), називається *задачею Больца*. В окремих випадках функціонал (1) може приймати такі види:

$$J = g_0(X(t_0), X(t_\varepsilon), t_0, t_\varepsilon). \quad (2.14)$$

$$J = \int_{t_0}^{t_\varepsilon} f_0(X(t), U(t), t) dt. \quad (2.15)$$

У першому випадку задача пошуку екстремуму називається *задачею Майера*, у другому – *задачею Лагранжа*.

Прикладами задачі Майера є: задача максимальної швидкодії

$$J = t_\varepsilon - t_0 \rightarrow \min, \quad (2.16)$$

задача на максимальну відстань переміщення

$$J = x(t_{\hat{e}}) - x(t_0) \rightarrow \max. \quad (2.17)$$

Приклад задачі Лагранжа - задача на мінімальне енергоспоживання:

$$J = \int_{t_0}^{t_{\hat{e}}} u^2(t) dt \rightarrow \min. \quad (2.18)$$

Вид підінтегральної функції критерію (2.18) пояснюється тим, що потужність керуючого сигналу, як правило, пропорційна квадрату його амплітуди. Крім того, використання квадрату, а не першого ступеня керування $u(t)$ дозволяє врахувати ту обставину, що в перехідному процесі керування може бути від'ємним. В окремих випадках, коли відомо, що керування завжди додатне, функціонал може бути й більш простим:

$$J = \int_{t_0}^{t_{\hat{e}}} u(t) dt \rightarrow \min. \quad (2.19)$$

Можна показати, що задачі Майера й Лагранжа мають однаковий ступінь загальності, тобто шляхом певних перетворень можна задачу Лагранжа представити у вигляді задачі Майера і навпаки.

Важливою обставиною при розв'язуванні задач оптимального керування є те, що компоненти векторів X і U не можуть розглядатися як незалежні функції часу, здатні приймати будь-які значення. На вектори X і U обов'язково накладаються деякі обмеження у вигляді рівнянь зв'язку, гранично припустимих значень тощо. Як мінімум, варто вказати на диференціальні рівняння самого об'єкта керування, що зв'язують компоненти векторів X , \dot{X} і U . Наприклад, розглянемо прямолінійний рух тіла масою m під дією керуючого впливу u , що створюється встановленим на тілі двигуном. Позначимо через x

координату центру мас тіла й припустимо, що ніякі інші сили на тіло не діють. Тоді у відповідності із другим законом Ньютона рівняння руху тіла має вигляд $\ddot{x}(t)m = u$. Останнє рівняння еквівалентне системі двох рівнянь першого порядку: $\dot{x}_1 = x_2$, $\dot{x}_2 m = u$.

Розділяють „класичні” (у вигляді рівностей) і „некласичні” (у вигляді нерівностей) обмеження. „Класичні”, у свою чергу, діляться на *голономні*, *неголономні* й *ізопериметричні*.

Голономні обмеження являють собою алгебраїчні рівняння зв'язку шуканих функцій $X(t)$ і $U(t)$, записані, для зручності, у вигляді однорідних рівнянь із нульовою правою частиною:

$$\varphi_i(X, U, t) = 0, \quad i = 1 \dots r, \quad (2.20)$$

де r - кількість алгебраїчних рівнянь.

Для задач оптимізації динамічних режимів роботи об'єктів керування голономні обмеження нетипові. Крім того, як правило, цих обмежень можна позбутися ще на етапі формулювання задачі шляхом відповідних перетворень. Тому надалі вони не розглядаються.

Неголономні обмеження являють собою диференціальні рівняння:

$$\varphi_i(X, \dot{X}, U, t) = 0, \quad i = 1 \dots n, \quad (2.21)$$

де n - кількість диференціальних рівнянь.

Це диференціальні рівняння об'єкта керування, а також інші рівняння, що дозволяють врахувати додаткові обмеження.

Ізопериметричні обмеження мають вигляд:

$$\int_{t_0}^{t_g} \varphi_i(X, U, t) dt = c_i = const, \quad i = 1 \dots k, \quad (2.22)$$

де k - кількість інтегральних рівнянь.

Як приклад такого обмеження можна привести обмеження на витрату енергії в перехідному процесі:

$$\int_{t_0}^{t_\epsilon} u^2(t) dt = c = \text{const.} \quad (2.23)$$

За допомогою певних перетворень ізопериметричні обмеження перетворюються в неголономні. Це перетворення полягає у введенні додаткових змінних, похідні яких за часом рівні підінтегральним виразам (2.22):

$$\dot{x}_{n+i} = \varphi_i(X, U, t), \quad i = 1 \dots z, \quad (2.24)$$

де z - кількість „нових” додаткових умов.

Умовно говорячи, нові змінні „розширюють” вихідну систему рівнянь об’єкта. Підставляючи (2.24) в (2.22), одержимо:

$$\int_{t_0}^{t_\epsilon} \varphi_i(X, U, t) dt = \int_{t_0}^{t_\epsilon} \dot{x}_{n+i} dt = x_{n+i}(t_\epsilon) - x_{n+i}(t_0) = c_i. \quad (2.25)$$

Для спрощення вважають $x_{n+i}(t_0) = 0$, тоді $x_{n+i}(t_\epsilon) = c_i$.

Типовим прикладом некласичних обмежень є обмеження на максимальні значення керуючих величин (обмеження на керування по модулю):

$$|u_i| \leq u_{i,\max}, \quad i = 1..m, \quad (2.26)$$

де m - кількість обмежень на керування.

Інший вид додаткових умов, що накладаються на задачу – це *крайові умови*, що визначають значення змінних об’єкта в початковий і кінцевий моменти часу перехідного процесу. За видом крайових умов розрізняють *задачі*

із закріпленими кінцями, коли $X(t_0)$ і $X(t_k)$ відомі (задані), і *задачі з рухомими кінцями*, коли частина або всі компоненти цих векторів невідомі (можуть приймати довільні значення). Серед останніх задач часто зустрічаються *задачі з вільним правим кінцем*, у якій вектор $X(t_k)$ невідомий.

Залежно від визначеності моменту часу t_k задачі розділяють на *задачі з фіксованим і нефіксованим часом*. До останнього типу задачі відноситься задачі на максимальну швидкодію.

Отже, задача оптимізації керування полягає в тому, щоб знайти такі вектори $U(t)$ і $X(t)$, які доставляють екстремум функціоналу критерію оптимальності з урахуванням усіх обмежень і крайових умов. Ці вектори називаються відповідно оптимальним керуванням і оптимальною траєкторією. У результаті розв'язку задачі оптимальне керування може бути знайдене або як *оптимальна програма*

$$U = U(t), \quad (2.27)$$

або як *оптимальна стратегія*

$$U = U(X). \quad (2.28)$$

Для побудови системи керування другий розв'язок, мабуть, більш бажаний, тому що дозволяє побудувати замкнену систему, здатну оптимальним чином функціонувати при будь-яких початкових умовах. Однак визначити оптимальну стратегію, як правило, набагато складніше, чим оптимальну програму.

Розв'язування задач оптимізації динамічних режимів здійснюється різними методами, основними з яких є: класичне варіаційне числення; метод максимуму Понтрягіна; динамічне програмування Беллмана [45].

2.5.3. Класичне варіаційне числення

2.5.3.1. Задача із закріпленими кінцями й фіксованим часом

Розглянемо задачу пошуку безумовного екстремуму функціонала виду:

$$J(y(t), \dot{y}(t), t) = \int_{t_0}^{t_e} f_0(y(t), \dot{y}(t), t) dt, \quad (2.29)$$

де $y(t)$ - деяка функція незалежної змінної t (надалі - часу).

Припустимо, що функція $y(t)$ доставляє екстремум функціоналу (2.29).

Дамо цій функції приріст $\Delta y(t)$ такий, що:

$$\begin{cases} \Delta y(t_0) = 0, \\ \Delta y(t_e) = 0. \end{cases} \quad (2.30)$$

Визначимо збільшення функціонала (2.29), викликане приростом функції $\Delta y(t)$:

$$\begin{aligned} \Delta J &= J(y(t) + \Delta y(t)) - J(y(t)) = \\ &= \int_{t_0}^{t_e} \left[f_0 \left(y(t) + \Delta y(t), \dot{y}(t) + \frac{d\Delta y}{dt}, t \right) - f_0(y(t), \dot{y}(t), t) \right] dt. \end{aligned} \quad (2.31)$$

Розклавши підінтегральний вираз (2.31) у ряд Тейлора й відкинувши всі доданки вище першого порядку малості, одержимо так звану першу варіацію функціонала:

$$\delta J = \int_{t_0}^{t_e} \left[\frac{\partial f_0}{\partial y} \Delta y(t) + \frac{\partial f_0}{\partial \dot{y}} \frac{d\Delta y}{dt} \right] dt. \quad (2.32)$$

Перша варіація функціонала є головною (лінійною) частиною його збільшення. Проінтегруємо другий доданок підінтегрального виразу (2.32) по частинах:

$$\int_{t_0}^{t_e} \frac{\partial f_0}{\partial \dot{y}} \frac{d\Delta y}{dt} dt = \frac{\partial f_0}{\partial \dot{y}} \Delta y \Big|_{t_0}^{t_e} - \int_{t_0}^{t_e} \frac{d}{dt} \frac{\partial f_0}{\partial \dot{y}} \Delta y(t) dt. \quad (2.33)$$

Беручи до уваги вираз (2.30), одержимо:

$$\int_{t_0}^{t_e} \frac{\partial f_0}{\partial \dot{y}} \frac{d\Delta y}{dt} dt = - \int_{t_0}^{t_e} \frac{d}{dt} \frac{\partial f_0}{\partial \dot{y}} \Delta y(t) dt. \quad (2.34)$$

Підставляючи вираз (2.34) в (2.32), запишемо:

$$\delta J = \int_{t_0}^{t_e} \left[\frac{\partial f_0}{\partial y} + \frac{d}{dt} \frac{\partial f_0}{\partial \dot{y}} \right] \Delta y(t) dt. \quad (2.35)$$

Оскільки функція $y(t)$ доставляє екстремум функціоналу (2.29), його перша варіація, визначена на цій функції, повинна бути рівною нулю:

$$\delta J = \int_{t_0}^{t_e} \left[\frac{\partial f_0}{\partial y} + \frac{d}{dt} \frac{\partial f_0}{\partial \dot{y}} \right] \Delta y(t) dt = 0. \quad (2.36)$$

Рівність (2.36) повинна мати місце для довільних функцій $\Delta y(t)$, що задовольняють крайовим умовам (2.30). Це можливо, якщо функція $\Delta y(t)$ множиться на нуль, тобто виконується рівність:

$$\frac{\partial f_0}{\partial y} - \frac{d}{dt} \frac{\partial f_0}{\partial \dot{y}} = 0. \quad (2.37)$$

Рівняння (2.37) називається *рівнянням Ейлера*.

Таким чином, якщо існує така функція $y(t)$, для якої функціонал (2.29) досягає екстремуму, то вона задовольняє рівнянню Ейлера (2.37).

Відзначимо, що рівняння (2.37) - це необхідна, але не достатня умова екстремуму функціонала (2.29), подібно тому, як рівність нулю похідної деякої

функції в деякій точці не є достатньою умовою екстремуму функції в цій точці. Більше того, якщо екстремум і досягається, то невідомий його вид (мінімум або максимум). Отже, розв'язок рівняння (2.37) потребує перевірки. Однак у багатьох випадках при розв'язуванні реальних задач оптимального керування отриманий розв'язок і вид екстремуму можна обґрунтувати фізичними міркуваннями і таку перевірку не виконують.

У випадку, якщо функціонал залежить від декількох функцій:

$$J = \int_{t_0}^{t_e} f_0(y_1(t), \dot{y}_1(t), y_2(t), \dot{y}_2(t), \dots, y_n(t), \dot{y}_n(t)) dt, \quad (2.38)$$

необхідними умовами його екстремуму будуть рівняння Ейлера, записані щодо всіх невідомих функцій:

$$\frac{\partial f}{\partial y_i} - \frac{d}{dt} \frac{\partial f}{\partial \dot{y}_i} = 0, \quad i = 1 \dots n, \quad (2.39)$$

де n – кількість невідомих функцій.

При розв'язуванні задач оптимального керування під функціями $y_i(t)$ потрібно розуміти невідомі траєкторії руху об'єкта $X(t)$ і саме керування $U(t)$. Тому в загальному випадку функціонал може мати вигляд:

$$J = \int_{t_0}^{t_e} f_0(x_1(t), \dot{x}_1(t), \dots, x_n(t), \dot{x}_n(t), u_1(t), \dot{u}_1(t), \dots, u_m(t), \dot{u}_m(t)) dt, \quad (2.40)$$

де m – кількість керувань об'єктом.

Однак похідні компонентів вектора стану X можуть бути виражені через самі компоненти й керуючі впливи за допомогою рівнянь об'єкта, а похідні вектора керування у функціонал у більшості випадків не входять. Крім того, надалі для спрощення будемо вважати керування скалярним. З врахуванням цього функціонал запишемо у вигляді:

$$J = \int_{t_0}^{t_e} f_0(x_1(t), \dots, x_n(t), u(t)) dt. \quad (2.41)$$

Таким чином, розглядається задача Лагранжа.

На змінні стану й керування накладені різні обмеження. У класичному варіаційному численні розглядаються обмеження у вигляді неперервних функцій. Стосовно задач оптимізації динамічних режимів ці функції зв'язують між собою змінні стану об'єкта й керуючий вплив і являють собою не що інше, як диференціальні рівняння об'єкта:

$$\dot{x}_i = f_i(X, u, t), \quad i = 1 \dots k, \quad (2.42)$$

де k – кількість диференціальних рівнянь руху об'єкта.

Рівняння (2.42) можуть бути представлені у вигляді неголономних обмежень у такий спосіб:

$$\varphi_i(X, \dot{X}, u, t) = f_i(X, u, t) - \dot{x}_i = 0. \quad (2.43)$$

Для врахування обмежень використовується метод невизначених множників Лагранжа. Згідно з ним підінтегральна функція критерію розширюється шляхом включення в неї обмежень (у нашому випадку – виду (2.43)). Отримана в такий спосіб функція називається *функцією Лагранжа* (*лагранжианом*):

$$L = f_0(X, u(t)) + \sum_{i=1}^n \psi_i(t) \varphi_i(X, \dot{X}, u, t), \quad (2.44)$$

де $\psi_i(t)$ – невідомі функції, що називаються множниками Лагранжа.

Відзначимо, що чисельно лагранжиан дорівнює функції $f_0(X, u(t))$.

Надалі при пошуку екстремуму замість вихідного функціонала використовується функціонал:

$$J^* = \int_{t_0}^{t_f} L(\dots) dt. \quad (2.45)$$

Рівняння Ейлера для функціонала (2.45) приймуть вид:

$$\frac{\partial J^*}{\partial x_i} - \frac{d}{dt} \frac{\partial J^*}{\partial \dot{x}_i} = 0, \quad (2.46)$$

$$\frac{\partial J^*}{\partial u} = 0. \quad (2.47)$$

Останнє рівняння являє собою рівняння Ейлера для керування. Його простий вид випливає з того, що в лагранжиан не входить похідна керування й

$$\frac{\partial J^*}{\partial \dot{u}} = 0.$$

Очевидні також наступні рівності:

$$\frac{\partial L}{\partial \psi_i} = 0, \quad (2.48)$$

що представляють собою неголономні обмеження.

Система рівнянь (2.46–2.48) називається *рівняннями Ейлера-Лагранжа*.

На практиці рівняння Ейлера-Лагранжа зручніше записувати через так звану *функцію Гамільтона (гамільтоніан)*:

$$H = f_0(X, u(t)) + \sum_{i=1}^n \psi_i(t) f_i(X, u, t) = L + \sum_{i=1}^n \psi_i(t) \dot{x}_i. \quad (2.49)$$

З врахуванням того, що $L = H - \sum_{i=1}^n \psi_i(t) \dot{x}_i$, $\frac{\partial L}{\partial \dot{x}_i} = -\psi_i$, рівняння (2.46-2.48)

приймуть вид:

$$\frac{\partial \dot{I}}{\partial x_i} + \frac{d\psi_i}{dt} = 0 \quad (2.50)$$

або

$$\dot{\psi}_i = -\frac{\partial \dot{I}}{\partial x_i}, \quad (2.51)$$

$$\frac{\partial \dot{I}}{\partial u} = 0. \quad (2.52)$$

Очевидно також, що:

$$\frac{\partial H}{\partial \psi_i} = f_i(X, u, t). \quad (2.53)$$

Рівняння (2.53) - це рівняння об'єкта, виражені через гамільтоніан. Надалі замість них можна безпосередньо використовувати вираз (2.42).

Система рівнянь (2.51)-(2.53) називається *системою Гамільтона*. Вона являє собою $2n$ звичайних диференціальних рівнянь першого порядку у формі Коші (рівняння (2.51) і рівняння об'єкта (2.53)).

Таким чином, якщо існує керування $u(t)$ і траєкторії $X(t)$, такі, що на них досягається екстремум функціонала (2.41), то снують не рівні одночасно нулю множники $\psi_i(t)$, що задовольняють рівнянням (2.51)–(2.53).

Щоб знайти оптимальне керування й оптимальні траєкторії необхідно розв'язати систему Гамільтона. Для цього, як відомо, потрібно $2n$ додаткових умов. Якщо розглядається задача з фіксованим часом і закріпленими кінцями, то

в якості додаткових виступають крайові умови: $X(t_0)=X_0$, $X(t_k)=X_k$ (кількість цих умов $2n$).

Виконаємо ще одне узагальнення: якщо функціонал (2.29) у підінтегральному виразі містить вищі похідні компонентів вектору стану X об'єкта керування, то необхідно використовувати рівняння Ейлера-Пуассона:

$$\sum_{i=0}^n (-1)^i \frac{d^i}{dt^i} \frac{\partial f_0(t, x, \dot{x}, \dots, x^{(n)})}{\partial x^i} = 0. \quad (2.54)$$

У виразі (2.54) прийнято, що найвища похідна, яка входить у функціонал (2.29) має порядок n [45].

2.5.3.2. Задача з незакріпленими кінцями й фіксованим часом

Як і раніше, розглянемо спочатку задачу пошуку безумовного екстремуму функціонала однієї функції:

$$J(y(t), \dot{y}(t), t) = g_0(y(t_0), y(t_\varepsilon)) + \int_{t_0}^{t_\varepsilon} f_0(y(t), \dot{y}(t), t) dt. \quad (2.55)$$

Інтегральна складова критерію визначає „якість” перехідного процесу на проміжку часу $[t_0; t_k]$. Функція $g_0(\dots)$ визначає складову якості, пов'язану з незакріпленими лівим і правим кінцями.

Відповідно до раніше наведеної класифікації задача пошуку екстремуму функціонала (2.55) є задачею Больца.

Для визначення необхідних умов екстремуму функціонала (43) необхідно знайти його першу варіацію й прирівняти її до нуля. Відзначимо, що збільшення $\Delta y(t)$ у цьому випадку приводить до збільшення інтеграла й функції $g_0(\dots)$. Це пов'язано зі збільшеннями значень функції Δy у незакріплених кінцях:

$$\begin{cases} \Delta y(t_0) \neq 0, \\ \Delta y(t_\epsilon) \neq 0. \end{cases} \quad (2.56)$$

Визначимо збільшення функціонала (43), викликане збільшенням функції $\Delta y(t)$:

$$\begin{aligned} \Delta J &= g_0(y(t_0) + \Delta y(t_0), y(t_\epsilon) + \Delta y(t_\epsilon)) - g_0(y(t_0), y(t_\epsilon)) + \\ &+ \int_{t_0}^{t_\epsilon} \left[f_0\left(y(t) + \Delta y(t), \dot{y}(t) + \frac{d\Delta y}{dt}, t\right) - f_0(y(t), \dot{y}(t), t) \right] dt. \end{aligned} \quad (2.57)$$

Розклавши ΔJ в ряд Тейлора й відкинувши всі доданки вищих порядків малості, одержимо першу варіацію функціонала:

$$\delta J = \frac{\partial g_0}{\partial y(t_0)} \Delta y(t_0) + \frac{\partial g_0}{\partial y(t_\epsilon)} \Delta y(t_\epsilon) + \int_{t_0}^{t_\epsilon} \left[\frac{\partial f_0}{\partial y} \Delta y(t) + \frac{\partial f_0}{\partial \dot{y}} \frac{d\Delta y}{dt} \right] dt. \quad (2.58)$$

Проінтегрувавши другий доданок підінтегрального виразу (2.58) по частинах, одержимо:

$$\delta J = \frac{\partial g_0}{\partial y(t_0)} \Delta y(t_0) + \frac{\partial g_0}{\partial y(t_\epsilon)} \Delta y(t_\epsilon) + \int_{t_0}^{t_\epsilon} \frac{\partial f_0}{\partial y} \Delta y(t) dt + \frac{\partial f_0}{\partial \dot{y}} \Delta y \Big|_{t_0}^{t_\epsilon} - \int_{t_0}^{t_\epsilon} \frac{d}{dt} \frac{\partial f_0}{\partial \dot{y}} \Delta y(t) dt. \quad (2.59)$$

Оскільки функція $y(t)$ доставляє екстремум функціоналу (2.55), його перша варіація на цій функції, повинна бути рівною нулю. Згрупувавши доданки в (2.59), запишемо:

$$\begin{aligned} \delta J &= \left[\frac{\partial g_0}{\partial y(t_0)} - \frac{\partial f_0}{\partial \dot{y}}(t_0) \right] \Delta y(t_0) + \left[\frac{\partial g_0}{\partial y(t_\epsilon)} + \frac{\partial f_0}{\partial \dot{y}}(t_\epsilon) \right] \Delta y(t_\epsilon) + \\ &+ \int_{t_0}^{t_\epsilon} \left[\frac{\partial f_0}{\partial y} - \frac{d}{dt} \frac{\partial f_0}{\partial \dot{y}} \right] \Delta y(t) dt = 0. \end{aligned} \quad (2.60)$$

Рівність (2.60) повинна виконуватися при будь-яких функціях $\Delta y(t)$ і їх крайових значеннях $\Delta y(t_0)$ і $\Delta y(t_k)$. Тому необхідні умови екстремуму функціонала (2.55) можна записати у вигляді:

$$\frac{\partial g_0}{\partial y(t_0)} - \frac{\partial f_0}{\partial \dot{y}}(t_0) = 0, \quad (2.61)$$

$$\frac{\partial g_0}{\partial y(t_e)} + \frac{\partial f_0}{\partial \dot{y}}(t_e) = 0, \quad (2.62)$$

$$\frac{\partial f_0}{\partial y} - \frac{d}{dt} \frac{\partial f_0}{\partial \dot{y}} = 0. \quad (2.63)$$

Рівняння (2.63) є рівняння Ейлера, а рівняння (2.61), (2.62) називаються *умовами трансверсальності*.

Умови трансверсальності зв'язують частинні похідні функцій $g_0(\dots)$ і $f_0(\dots)$ і після їхнього визначення перетворюються в алгебраїчні рівняння.

У випадку, коли функціонал залежить від декількох функцій, рівняння Ейлера записуються для всіх функцій, а умови трансверсальності - для всіх незакріплених кінців.

Стосовно задач оптимального керування об'єктом, що описується n диференціальними рівняннями першого порядку, функціонал якості приводиться звичайно до виду:

$$J = g_0(x_1(t_0), \dots, x_n(t_0), x_1(t_e), \dots, x_n(t_e)) + \int_{t_0}^{t_e} f_0(x_1(t), \dots, x_n(t), u(t), t) dt. \quad (2.64)$$

Ми маємо n незакріплених лівих і n незакріплених правих кінців, що показано в (2.64). На практиці число незакріплених кінців звичайно менше, і $g_0(\dots)$ є функцією менш ніж $2n$ змінних.

Для врахування обмежень, що накладаються на функції $x_i(t)$, керування $u(t)$ і незакріплені кінці, використовується метод невизначених множників Лагранжа. Вихідний функціонал (2.64) „розширюється” шляхом включення в нього всіх обмежень (у вигляді лівих частин рівностей), помножених на невизначені множники Лагранжа:

$$J^* = G(\dots) + \int_{t_0}^{t_e} L(\dots) dt, \quad (2.65)$$

$$G = g_0(\dots) + \sum_{i=1}^r v_i g_i(\dots), \quad (2.66)$$

$$L = f_0(X, u(t)) + \sum_{i=1}^n \psi_i(t) \varphi_i(X, \dot{X}, u, t), \quad (2.67)$$

де $g_i(\dots)=0$ – обмеження, що накладаються на незакріплені кінці (максимальна їхня кількість дорівнює числу незакріплених кінців); v_i – невизначені множники Лагранжа.

Необхідними умовами екстремуму функціонала (2.65) будуть:

$$\begin{cases} \frac{\partial L}{\partial x_i} - \frac{d}{dt} \frac{\partial L}{\partial \dot{x}_i} = 0, \\ \frac{\partial L}{\partial u} = 0, \end{cases} \quad (2.68)$$

$$\frac{\partial G}{\partial x_j(t_0)} - \frac{\partial L}{\partial \dot{x}_j} \Big|_{t_0} = 0, \quad (2.69)$$

де j - індекс незакріплених лівих кінців,

$$\frac{\partial G}{\partial x_s(t_e)} + \frac{\partial L}{\partial \dot{x}_s} \Big|_{t_e} = 0, \quad (2.70)$$

де s - індекс незакріплених правих кінців.

Рівняння (2.68) є рівняння Ейлера-Лагранжа, а рівняння (2.69) і (2.70) - умовами трансверсальності для функціонала (2.65).

Враховуючи, що $L = H - \sum_{i=1}^n \psi_i(t)\dot{x}_i$, рівняння (2.68)–(2.70) можна записати через функцію Гамільтона:

$$\begin{cases} \frac{\partial H}{\partial x_i} = -\dot{\psi}_i, \\ \frac{\partial H}{\partial u} = 0, \end{cases} \quad (2.71)$$

$$\frac{\partial G}{\partial x_j(t_0)} = -\psi_j(t_0), \quad (2.72)$$

$$\frac{\partial G}{\partial x_s(t_e)} = \psi_s(t_e). \quad (2.73)$$

Як вже вказувалося, рівняння (2.71) разом з рівняннями об'єкта утворюють систему з $2n$ диференціальних рівнянь першого порядку. Для їхнього розв'язку необхідно $2n$ додаткових умов. Якщо задача містить r незакріплених кінців, то ми маємо $2n$ -г крайових умов і r умов трансверсальності (2.72), (2.73). Таким чином, загальне число умов рівне $2n$, і, отже, задача має розв'язок [45].

2.5.4. Принцип максимуму Понтрягіна

Принципом максимуму називають математичний метод, розроблений академіком Л.С. Понтрягіним і його учнями для розв'язування задач оптимального керування. Цей метод дозволяє визначити необхідні умови екстремуму функціонала виду:

$$J = \int_{t_0}^{t_e} f_0(X(t), U(t)) dt, \quad (2.74)$$

при наявності будь-яких, у тому числі неklasичних (у вигляді нерівностей) обмежень, що накладаються на вектори X і U .

Важливо відзначити, що при цьому кінцевий час t_k невідомий й підлягає визначенню при розв'язуванні задачі.

Ми будемо розглядати задачі оптимального керування, коли неklasичні обмеження задані тільки на вектор керування, чому відповідає варіант принципу максимуму, який найбільш часто використовується на практиці. Будемо вважати, що цей вектор повинен належати деякій відомій області припустимих значень Ω , тобто $U \subset \Omega$. Область може бути задана, наприклад, нерівностями виду (2.26). Для визначення необхідних умов екстремуму функціонала (2.74) принцип максимуму, як і варіаційне числення, використовує функцію Гамільтона. Ця функція задається в такий спосіб:

$$H = \psi_0(t)f_0(X(t), U(t)) + \sum_{i=1}^n \psi_i(t)f_i(X(t), U(t)), \quad (2.75)$$

де $\psi_0 \dots \psi_n$ – невизначені множники; $f_i(X(t), U(t))$ – праві частини диференціальних рівнянь об'єкта представлені у формі Коші.

Відмінністю даної функції від гамільтоніана, що використовується у варіаційному численні, є присутність множника ψ_0 перед підінтегральною функцією функціонала.

При фіксованих значеннях $\psi_0 \dots \psi_n$ і $x_1 \dots x_n$ функція H є функцією керування U . Отже, з області припустимих значень можна вибрати такий вектор керуючих впливів, при якому гамільтоніан досягає максимуму. Позначимо верхню грань (максимум) значень H через $M(\Psi, X)$:

$$M(\Psi, X) = \sup_{U \subset \Omega} H(\Psi, X, U). \quad (2.76)$$

Сформулюємо принцип максимуму.

Для оптимальності керування $u(t)$ й траєкторії $X(t)$ необхідно існування такої ненульової неперервної вектор-функції $\Psi(t)=[\psi_0(t), \psi_1(t), \dots, \psi_n(t)]$, яка відповідає функціям $U(t)$ і $X(t)$, що:

- 1) при будь-якому t на відрізку $t_0 \leq t \leq t_\epsilon$ функція $H(\Psi, X, U)$ змінного $U \subset \Omega$ досягає при даному керуванні $U(t)$ максимуму:

$$H(\Psi(t), X(t), U(t)) = M(\Psi(t), X(t)); \quad (2.77)$$

- 2) у кінцевий момент часу t_ϵ виконуються співвідношення:

$$\psi_0(t_\epsilon) \leq 0; \quad \dot{I}(\Psi(t_\epsilon), X(t_\epsilon)) = 0. \quad (2.78)$$

Крім того, якщо $\Psi(t), X(t), U(t)$ задовольняють рівнянням:

$$\dot{\psi}_i = -\frac{\partial I}{\partial x_i}, \quad i = 1 \dots n \quad (2.79)$$

і умові (2.77), то функції $\psi_0(t)$ і $M(\Psi(t), X(t))$ змінного t є постійними й умова (2.78) виконується в будь-який момент часу на відрізку $t_0 \leq t \leq t_\epsilon$.

Відзначимо, що вектор $\Psi(t)$ може бути визначений з точністю до постійного множника, оскільки очевидно, що якщо деякий вектор $\Psi^*(t)$ задовольняє умовам принципу максимуму, то вектор $k\Psi^*(t)$, де $k=\text{const}>0$, також їм задовольняє. Тому завжди при $\psi_0(t) \neq 0$ можна покласти $\psi_0(t) = -1$, що й робиться на практиці (випадок $\psi_0(t) = 0$ є особливим і нами не розглядається) [45].

2.5.5. Метод динамічного програмування

Метод динамічного програмування запропонований Р. Беллманом в основному для оптимізації дискретних багатокрокових процесів, однак застосовується й для неперервних систем.

На відміну від варіаційного числення й принципу максимуму, спрямованих на відшукування оптимального керування у вигляді оптимальної програми, метод динамічного програмування орієнтований на пошук оптимальної стратегії.

В основу методу покладений наступний *принцип оптимальності*: *оптимальна стратегія має ту властивість, що незалежно від того, яким був первісний стан системи й первісний розв'язок (керування), наступні розв'язки (наступне керування) повинні бути оптимальні щодо стану, який виник після прийняття першого розв'язку*. Це означає, що оптимальна стратегія в будь-який момент часу визначається тільки тим станом, у якому перебуває система в даний момент.

Розглянемо застосування методу для неперервних систем. Для простоти будемо вважати керування скалярним. Нехай рух об'єкта визначається рівняннями

$$\dot{x}_i = f_i(x_1, x_2, \dots, x_n, u, t), \quad i = 1 \dots n \quad (2.80)$$

або у векторній формі:

$$\dot{X} = f(X, u, t). \quad (2.81)$$

Необхідно визначити оптимальну стратегію $u=u(X)$, яка мінімізує функціонал:

$$J = \int_0^{t_2} f_0(X(t), u(t)) dt \quad (2.82)$$

з урахуванням крайової умови $X(t_k) = X_k$.

Згідно з методом динамічного програмування за початковий стан об'єкта можна прийняти будь-який стан X . При цьому мінімальне значення функціонала й оптимальне керування, що переводить об'єкт зі стану X у стан X_k , однозначно визначаються станом X . Позначимо мінімальне значення функціонала так $S(X)$:

$$S(X) = \min_{u \in U} \int_0^{t_2} f_0(X(t), u(t)) dt, \quad (2.83)$$

де U - область припустимих керувань.

Відмітимо, що $S(X_k) = 0$, що безпосередньо випливає з (2.83). Інтеграл, що входить в вираз (2.83), можна представити у вигляді:

$$\int_0^{t_2} f_0(X(t), u(t)) dt = \int_0^{\Delta t} f_0(X(t), u(t)) dt + \int_{\Delta t}^{t_2} f_0(X(t), u(t)) dt. \quad (2.84)$$

Допустимо, на інтервалі $t = 0 \dots \Delta t$ керування було оптимальним. Надалі керування повинно вибиратися, виходячи із принципу оптимальності:

$$\int_{\Delta t}^{t_2} f_0(X(t), u(t)) dt = S(X(\Delta t)) \quad (2.85)$$

тобто повинно бути оптимальним щодо стану $X(\Delta t)$.

Оскільки траєкторія $X(t)$ неперервна, то при $\Delta t \rightarrow 0$ справедливо наступне:

$$X(\Delta t) \rightarrow X + \dot{X} \Delta t = X + f(X, u) \Delta t, \quad (2.86)$$

$$\int_0^{\Delta t} f_0(X(t), u(t)) dt \rightarrow f_0(X, u) \Delta t, \quad (2.87)$$

де X, u - значення координат об'єкта й керування в початковий момент часу.

Підставимо вираз (2.86) в (2.85). Результат разом з (2.87) підставимо в (2.84) і далі в (2.82), (2.83), будемо мати:

$$J = f_0(X, u) \Delta t + S(X + f(X, u) \Delta t), \quad (2.88)$$

$$S(X) = \min_{u \in U} (f_0(X, u) \Delta t + S(X + f(X, u) \Delta t)), \quad (2.89)$$

Припустимо, що S – неперервно-диференційована по X функція, тоді при $\Delta t \rightarrow 0$:

$$S(X + f(X, u) \Delta t) \rightarrow S(X) + \frac{\partial S}{\partial X} f(X, u) \Delta t, \quad (2.90)$$

де

$$\frac{\partial S}{\partial X} f(X, u) = \frac{\partial S}{\partial x_1} f_1(X, u) + \frac{\partial S}{\partial x_2} f_2(X, u) + \dots + \frac{\partial S}{\partial x_n} f_n(X, u). \quad (2.91)$$

Отже:

$$S(X) = \min_{u \in U} \left(f_0(X, u) \Delta t + S(X) + \frac{\partial S}{\partial X} f(X, u) \Delta t \right). \quad (2.92)$$

З виразу (2.91) одержимо:

$$0 = \min_{u \in U} \left(f_0(X, u) \Delta t + \frac{\partial S}{\partial X} f(X, u) \Delta t \right). \quad (2.93)$$

Оскільки $\Delta t \neq 0$, остаточно запишемо:

$$0 = \min_{u \in U} \left(f_0(X, u) + \frac{\partial S}{\partial X} f(X, u) \right). \quad (2.94)$$

Рівняння (2.94) називається *функціональним рівнянням Беллмана*. Воно зв'язує $S(X)$ з функціями u і X , що забезпечують мінімум функціонала (2.82). Його розв'язок дозволяє визначити не тільки функцію $S(X)$, але й оптимальні траєкторії і оптимальне керування. Якщо функціонал, який потребує мінімізації є квадратичним, то розв'язок рівняння (2.94) прийнято шукати у вигляді квадратичної форми.

Зазначимо, що використання дискретного динамічного програмування дозволяє отримати глобальний мінімум функціоналу, однак при його знаходженні необхідна велика кількість пам'яті обчислювального комп'ютера (мікроконтролера). Причому збільшення кількості координат, по яких може рухатись мехатронна система, значно збільшує вимоги до об'єму пам'яті (прокляття розмірності).

Перевагою методу динамічного програмування є те, що воно дає змогу отримати адаптивне оптимальне керування, тобто таке, яке є функцією поточних координат мехатронної системи $u=u(t, x(t))$. Недоліком притаманним даному методу є складність (в деяких випадках неможливість) розв'язання рівняння Беллмана (2.94) [45].

2.5.6. Наближені методи синтезу оптимального керування

У зв'язку зі значними труднощами побудови аналітичного розв'язку задач оптимального керування виняткове значення набули різні наближені й чисельні методи їх дослідження. Залежно від алгоритмічної основи методу він може бути віднесений до тієї або іншої групи. Охарактеризуємо деякі з них. В основі

першої групи методів лежить можливість зведення задачі оптимального керування до крайової задачі для системи диференціальних рівнянь за допомогою принципу максимуму з наступним використанням різних алгоритмів задання відсутніх початкових умов. До другої групи чисельних методів відносяться ті, у яких шукається оптимальне керування за допомогою ітераційних процедур у просторі керувань. При цьому використовуються формули для збільшення критерію якості при варіації керування, що приводять або до методів градієнтного типу в просторі керувань, або до процедур послідовних наближень. Третю групу становлять чисельні процедури, засновані на переборі в просторі траєкторій або методи динамічного програмування. Крім того, є методи, ефективні для конкретних класів систем, наприклад для лінійних систем, а також методи, які вихідну задачу оптимального керування зводять до задачі математичного програмування [46].

У додатку 1 приведені приклади розв'язування задач оптимального керування.

2.6. Інтелектуальні мехатронні системи керування

2.6.1. Основні поняття штучого інтелекту. Аналогії з процесами мислення людини

Термін „штучний інтелект” уперше з'явився наприкінці 60-х рр. Штучний інтелект (англ. - artificial intelligence) – це штучні програмні системи, створені людиною на базі ЕОМ, що імітують вирішення людиною складних творчих завдань у процесі її життєдіяльності. Інше визначення: штучний інтелект – це програми для ЕОМ, за допомогою яких машина здобуває здатність вирішувати нетривіальні завдання й ставити нетривіальні запитання.

Розрізняють два напрямки робіт у дослідженні штучного інтелекту. Перший із цих напрямків, який можна умовно назвати біонічним, має своєю метою змоделювати діяльність мозку, його психофізіологічні властивості, щоб спробувати відтворити на ЕОМ або за допомогою спеціальних технічних пристроїв штучний розум (інтелект).

Другий (основний) напрямок робіт в області штучного інтелекту, який інколи називають прагматичним, пов'язаний зі створенням систем автоматичного розв'язку складних (творчих) завдань на ЕОМ без врахування природи тих процесів, які відбуваються в людській свідомості, при їх розв'язуванні. Створені в рамках цього напрямку системи штучного інтелекту дозволяють доводити математичні теореми, перекладати тексти з однієї мови на іншу, діагностувати хвороби, уміло грати в шахи й інші інтелектуальні ігри. Нове покоління роботів, наділених штучним інтелектом, має такі „інтелектуальні” властивості, як здатність навчатися, пристосовуватися до змін зовнішнього середовища, „осмислено” імітуючи поведінку людини.

Незважаючи на відмінності предметної області (тобто сфери застосування) систем штучного інтелекту, можна виділити деякі загальні принципи їх побудови й функціонування. Насамперед, відзначимо, що процес мислення людини має ряд характерних особливостей:

- існує **мета**, тобто той кінцевий результат, на який спрямовані розумові процеси людини („мета змушує людину думати”);
- людський мозок зберігає величезне число **фактів** і **правил** їх використання. Для досягнення певної мети треба тільки звернутися до потрібних фактів і правил;
- прийняття рішень завжди здійснюється на основі спеціального **механізму спрощення**, що дозволяє відкидати непотрібні (малоістотні) факти й правила, які не мають відношення до завдання, що розв'язується в цей

момент, і, навпаки, виділяти головні, найбільш значимі факти й правила, потрібні для досягнення мети;

- досягаючи мети, людина не тільки приходить до розв'язку поставленого перед ним завдання, але й одночасно здобуває нові знання. Та частина інтелекту, яка дозволяє йому робити відповідні висновки на підставі правил, відібраних механізмом спрощення, і генерувати нові факти із уже існуючих, називається **механізмом** (або машиною) **виводу** (inference machine).

Оскільки система штучного інтелекту приймає рішення аналогічно тому, як це робить людина, то вона повинна містити в собі наступні ключові елементи – мета, факти (вихідні дані), правила, механізми виводу і спрощення.

Одним із творців кібернетики Джоном фон Нейманом було помічено, що прагнення одержати точну, вичерпну модель для складного об'єкта (процесу) не має сенсу, оскільки складність такого опису стає порівнянною зі складністю самого об'єкта. Отже, використання складної моделі об'єкта керування не дозволяє просто й наочно пояснити механізм його функціонування та використати які-небудь стандартні математичні процедури для дослідження характеристик об'єкта для синтезу системи керування ним.

Відомий фахівець в області теорії систем професор факультету електротехніки та інформатики Каліфорнійського університету (м. Берклі, США) Лотфі А. Заде сформулював цю думку у вигляді так званого принципу несумісності. Згідно із цим принципом: „Чим складніша система, тем менша наша здатність дати точне і практично цінне судження про її поведінку”. Для систем, складність яких перевершує деякий граничний рівень, точність і практичний сенс її використання стають характеристиками, які виключають одна одну. У більшості випадків особи, що приймають рішення, не можуть формально уявити собі увесь процес. І справа тут не в тому, що вони погано

розуміють те, що роблять, а в тому, що невизначеність (нечіткість) лежить у самій природі прийняття рішень. Виражаючись словами Л. Заде „у більшості основних завдань, що розв’язуються людиною, не потрібна висока точність. Людський мозок використовує допустимість такої неточності, кодуючи інформацію, достатню для розв’язку елементами нечітких величин, які лише приблизно описують вихідні дані” [47 - 49]. Потік інформації, що надходить у мозок через органи зору, слуху, дотику звужується у „тонкий струмок” інформації, необхідної для розв’язку поставленого завдання з мінімальним ступенем точності. Здатність оперувати нечіткими величинами є однією з найцінніших якостей людського розуму. Отже, людському мисленню характерні підходи для яких точність, строгість і математичний формалізм не є чимось необхідним і в яких використовується методологічна схема, що допускає нечіткості й часткові істини.

Класична логіка розвивається з найдавніших часів. Її основоположником вважається Аристотель. Класична або булева логіка має один істотний недолік – з її допомогою неможливо описати асоціативне мислення людини, вона оперує тільки двома поняттями: ІСТИНА й НЕПРАВДА, без будь-яких проміжних значень. Усе це добре для обчислювальних машин, але спробуйте представити весь навколишній наш світ тільки в чорному й білому кольорі, виключивши з мови будь-які відповіді на запитання, крім ТА й НІ. У такій ситуації вам можна тільки поспівчувати. Розв’язати цю проблему може нечітка логіка.

2.6.2. Використання систем з нечіткою логікою

В 1983 р. японська фірма „Фуджі Електрик” реалізувала на основі нечітких алгоритмів керування установкою для обробки питної води. В 1987 р. запущена у виробництво система керування новим метро в м. Сендаї, близько Токіо, запропонована на аналогічних принципах фірмою „Хітачі”. В 1991 р. Японія експортувала більш ніж на 25 млрд. доларів в товари, у яких використовувалися компоненти нечіткої логіки. Це, у першу чергу, товари культурно-побутового

призначення – фотоапарати, відеокамери, пральні машини, холодильники, пилососи, мікрохвильові печі тощо. Таким чином, технологія, не помічена світом, у Японії перетворилася в одну із ключових технологій, що відразу привернуло до себе величезну увагу. Сьогодні багато провідних компаній США, Німеччини, Франції й ряду інших країн пропонують найрізноманітніші товари й системи з використанням принципів нечіткої логіки, освоюючи все нові й нові області її застосування. Разом з тим, за оцінками європейських учених, Японія значно випереджає в розвитку своїх найближчих конкурентів у цьому напрямку.

Останнім часом нечітка технологія завойовує усе більше прихильників серед розробників систем керування. Стрибок у розвитку нечітких систем керування не випадковий. Простота й дешевизна їх розробки змушує проектувальників усе частіше використовувати цю технологію.

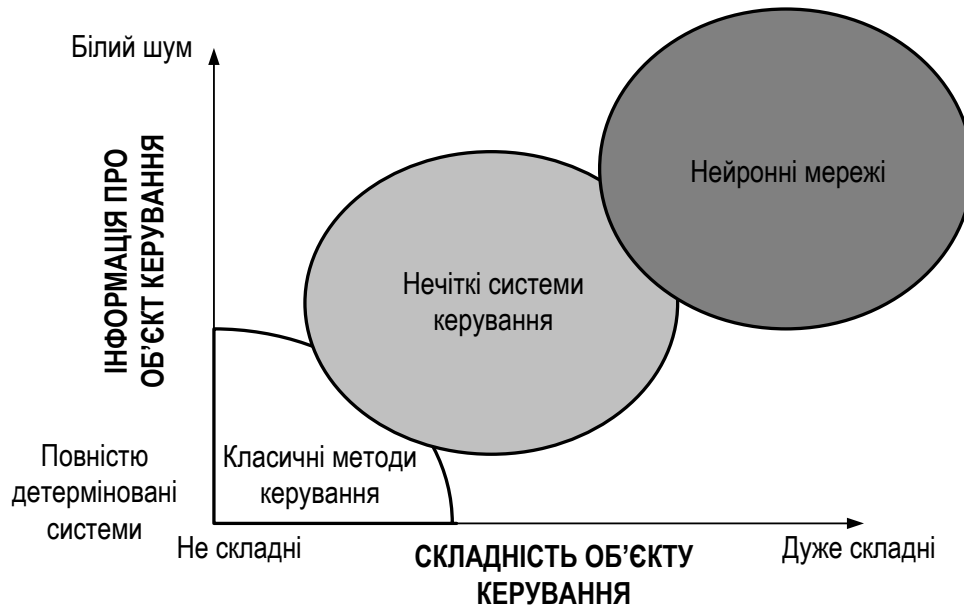


Рис. 2.15 Місце нечітких систем у сфері методів керування

На рис. 2.15 показані області найбільш ефективного застосування сучасних технологій керування. Як видно, класичні методи керування добре працюють при повністю детермінованому об'єкті керування й детермінованому середовищі, а для систем з неповною інформацією й високою складністю об'єкта керування

оптимальними є нечіткі методи керування. У правому верхньому куті малюнка наведена ще одна сучасна технологія керування – із застосуванням штучних нейронних мереж [50].

2.6.3. Основи нечіткої логіки (Fuzzy Logic)

Класична логіка із двома якісними або кількісними рівнями (істинно-неправильно, так - ні, одиниця - нуль) не дозволяє описати різноманіття світу й асоціативний (об'єднуючий у загальні поняття) спосіб мислення людини. Однак, нечітка логіка використовує багаторівневе представлення фізичних величин і тверджень. У класичній логіці фізичні величини й твердження діляться жорстко, безкомпромісно на два рівні. У фаззі-логіці крім крайніх двох рівнів є й проміжні, при цьому перехід від одного рівня до іншого не різкий, а з перекриттям сусідніх рівнів. Для певних фізичних величин і фактів їх дворівневе класичне представлення „правда-неправда”, „так-ні” повністю справедливо й точно. Наприклад, на запитання, чи рухається робочий орган робота, можна відповісти тільки однозначно „так” або „ні”. Але на запитання, як рухається робочий орган (швидко, повільно, дуже повільно), дворівнева логіка не дає відповідь, а фаззі-логіка дозволяє його одержати. Виділення рівнів у логіці - це асоціативна властивість мислення людини, коли якісна характеристика фізичних величин і процесів, формована словесно (прикметниками, прислівниками), оцінюється кількісно не окремими числами, а деякими числовими величинами з нерізкими границями.

Фундаментальним поняттям фаззі-логіки є фаззі-множина. У математичній логіці є загальне поняття множини як сукупності об'єктів, кожний з яких повинен мати або не мати певну властивість. Ця множина характеризується тільки одним показником, усі його об'єкти рівноцінні щодо визначальної його властивості. На відміну від даного поняття множини фаззі-множина характеризується двома показниками, по-перше, фактом

приналежності об'єктів до множини й, по-друге, ступенем їх принадлежности до даної множини. Стосовно до технічних систем об'єктами фазі-множини є значення деякої фізичної змінної, наприклад, значення температури, швидкості переміщення, електричної напруги, струму. Словесне (лінгвістичне) вираження фізичної змінної вважається логічною змінною у фазі-логіці. Властивістю, що поєднує значення фізичної змінної у фазі-множину, є деяка якісна оцінка в лінгвістичній формі для логічної змінної, наприклад, від'ємна мала (*NS*), нульова (*Z*), позитивна середня (*PM*), додатна велика (*PB*) тощо. Дані якісні оцінки фазі-множини називаються термами.

Ступінь принадлежности значень фізичної змінної до даного терму, визначається так званою функцією принадлежности $\mu(x)$. Значення функції принадлежности визначаються в межах від 0 до 1 на інтервалі від $x=a$, де $\mu(a)=0$, до $x=b$, де $\mu(b)=0$. У середині інтервалу $[a, b]$ є точка $x=c$, де $\mu(c)=1$ (рис. 2.16).

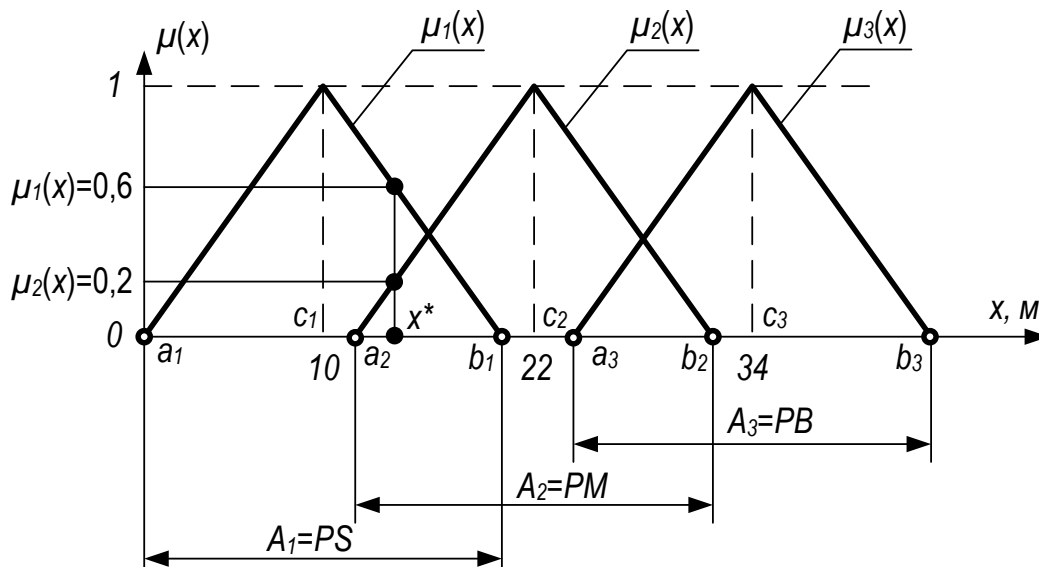


Рис. 2.16. Приклад фазі-множин та функцій принадлежности

На рис. 2.16, як приклад, змінна „відстань переміщення” представлена термами: додатна мала (*PS*) для $x = 0...20$ м; додатна середня (*PM*) для $x= 12...32$

м; додатна велика (PB) для $x= 24\dots44$ м. Виділеним фазовим множинам A_1, A_2, A_3 відповідають функції приналежності $\mu_1(x), \mu_2(x), \mu_3(x)$ із центрами c_1, c_2, c_3 , у яких $\mu(x)= 1$. Сама суть фаззі-логіки в її нетвердості й наближеності обумовлює наявність ділянок перекриття термів, де порушується однозначність приналежності значень змінної x тільки одному терму. Ширина ділянок перекриття може бути різною, навіть нульовою. Але максимальна ширина обмежується очевидною умовою: там, де для одного терма $\mu_i(x) = 1$ для іншого може бути тільки $1 > \mu_{i+1}(x) > 0$ та $1 > \mu_{i-1}(x) > 0$.

При побудові функцій приналежності пріоритетне значення набуває розміщення центрів функцій приналежності, де має місце $\mu_i(c_i)=1$, що означає безперечну (стоцентну) приналежність значення $x = c_i$, до відповідного терму. Якщо функція приналежності для відстані переміщення захвату робота складена експертом із використання робототехнічних систем, то відстань $x=22$ м задана за умовами технологічного процесу буде прийнята за середню, $x=10$ м - за малу, коли потрібне включення приводу захвату, а $x=34$ м - за велику, коли потрібне реверсування приводу захвату (захват робота вже „проїхав” точку адресації і необхідно його перемістити назад). При $x=x^*$ відстань переміщення відноситься одночасно до двох термів PS і PM , тому виникає проблема, яке із двох значень функції приналежності прийняти. Дана проблема вирішується компромісно на основі принципів сполучення множин, відомих з математичної логіки, у формі трьох операцій:

- ▶ усереднення, або кон'юнкції, $K(\mu_1, \mu_2) = \mu_1 \wedge \mu_2 = \min(\mu_1, \mu_2)$ - принцип мінімуму;
- ▶ об'єднання, або диз'юнкції, $D(\mu_1, \mu_2) = \mu_1 \vee \mu_2 = \max(\mu_1, \mu_2)$ - принцип максимуму;
- ▶ інверсії або доповнення, $\overline{\mu(x)} = 1 - \mu(x)$.

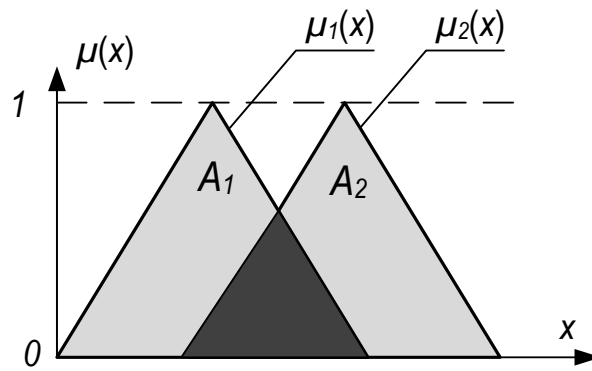


Рис. 2.17. Операції усереднення і об'єднання

Результат операцій усереднення й об'єднання фаззі-множин A_1 і A_2 однієї змінної x наведений на рис. 2.17. Сірий темний трикутник на рис. 2.17 означає кон'юнкцію двох функцій приналежності (μ_1 і μ_2), а фігура, яка утворюється із обох сірих фігур - диз'юнкцію двох функцій приналежності (μ_1 і μ_2). Аналогічні операції можуть бути використані й для більшого числа функцій приналежності. Наведені вище операції дозволяють прийняти одне певне значення функції приналежності в зоні взаємного перекриття фаззі-множин. Це дає підставу для прийняття певного рішення щодо значення вихідної змінної, тобто синтезу керуючого сигналу [22].

2.6.4. Структура й алгоритм фаззі-керування

Яка загальна структура фаззі-керування? Як і в будь-якій традиційній системі, у її складі можна виділити систему керування і об'єкт керування. Сама система фаззі-керування складається із чотирьох блоків, що виконують послідовно (в три етапи) процедуру формування алгоритму керування як функції керуючих впливів у від вхідних змінних $x_1, x_2, x_3, \dots, x_n$ (рис. 2.18). Блок фаззифікації (**Ff**) перетворює вхідні фізичні змінні ($x_1, x_2, x_3, \dots, x_n$) у терми A_i лінгвістичних змінних і виділяє для кожного моменту часу (для кожного часового такту) значення функції приналежності $\mu(x)$ для активізованих правил.

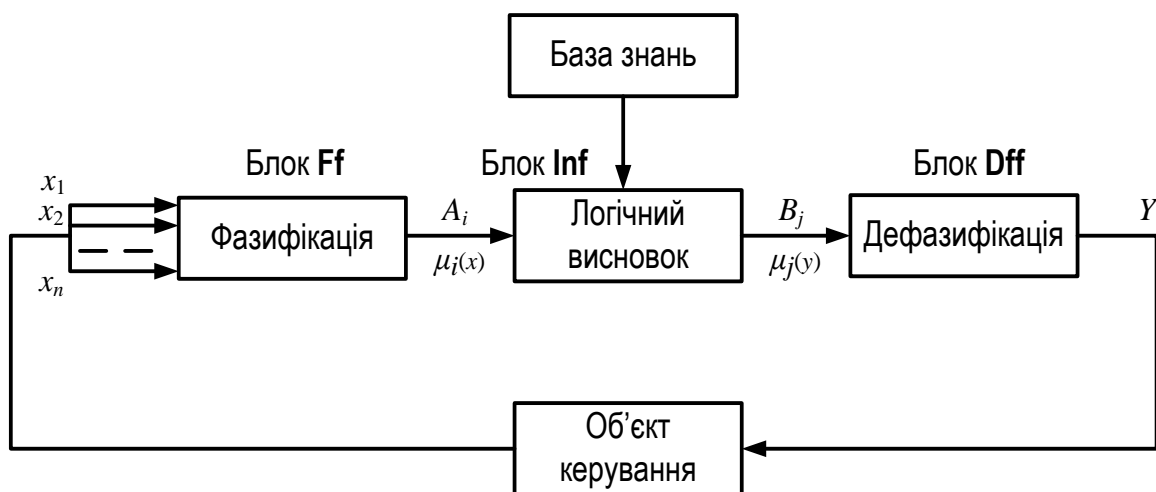


Рис. 2.18. Блочна структура системи фаззі-керування

Блок логічного висновку (**Inf**) відповідно до правил, закладених попередньо експертом в блок бази знань, визначає терми B_j вихідних лінгвістичних змінних й привласнює їм згідно із принципами фаззі-логіки значення функції приналежності $\mu_j(y)$.

Блок дефазифікації (**Dff**) перетворює терми B_j у керуючий сигнал Y . Якщо фаззі-пристрій повинен виробляти кілька керуючих впливів, то такий багатоканальний фаззі-пристрій можна розглядати як декілька паралельно працюючих пристроїв.

Основу алгоритму фаззі-керування становить масив правил, який перебуває в блоці бази знань. Правила складаються експертом на підставі його розуміння завдання керування. Якщо мова йде про регульовану динамічну систему зі складним або неповним математичним описом, то за основу алгоритму керування може бути, наприклад, прийнята умова виконання системою заданого руху з мінімальним відхиленням від заданої швидкості.

Якщо об'єкт керування має повний математичний опис, то за основу може бути прийнятий знайдений лінійний алгоритм, що вирішує завдання керування об'єктом. Відповідно до знайденого алгоритму правила перевіряються

(експериментально або за допомогою розрахунку) і при необхідності корегуються. Правила складаються з урахуванням наступних рекомендацій:

- ▶ масив правил повинен бути досить повним для виконання поставленого завдання керування, не повинно бути неврахованих ситуацій з вимірюваними координатами, що можуть призвести до втрати контролю над рухом;
- ▶ бажано, щоб кожне правило було простим, містило одну умову й один висновок;
- ▶ для одержання більш якісного процесу керування не повинно залишатися значень x_i не перекритих хоча б двома правилами фаззі-множин;
- ▶ чим різноманітніші тестові сигнали використовуються в процесі складання алгоритму, тим менш чутливу до появи непередбачуваних впливів можна одержати в умовах її експлуатації.

Додатково до масиву правил у склад алгоритму входять функції приналежності, що визначають кількісний взаємозв'язок фізичних змінних з лінгвістичними, тобто з термами. До форми функцій приналежності $\mu(x)$ пред'являються мінімальні вимоги в силу наближеного принципу керування на основі фаззі-логіки. Зараз сформувалося поняття про так звані стандартні функції приналежності (рис. 2.19).

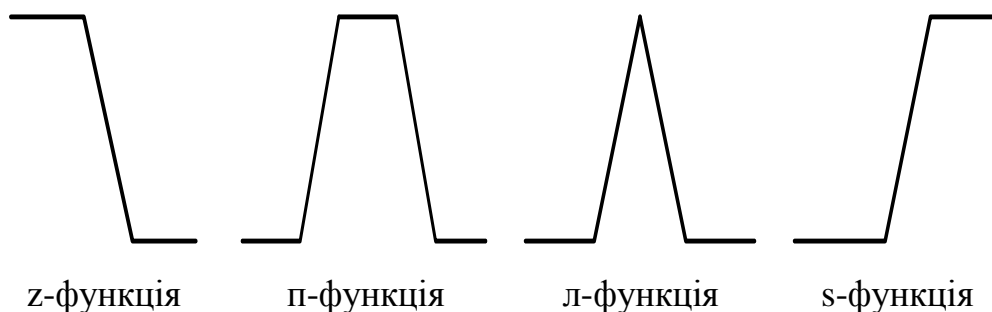


Рис. 2.19. Стандартні функції приналежності

Стандартні функції приналежності легко застосувати для розв'язку більшості завдань. Однак, якщо вирішується специфічне завдання, можна вибрати

й більш підходящу форму функції приналежності, при цьому можна добитися кращих результатів роботи системи.

Число функцій приналежності, що припадають на кожну фізичну змінну, вибирають із міркувань якості керування. Зі збільшенням числа функцій приналежності якість керування підвищується, однак при цьому ускладнюється алгоритм і підвищуються вимоги до його швидкодії. Наявний досвід побудови систем з фаззі-керуванням показав, що трьох-п'яти функцій приналежності для кожної вхідний змінної й п'яти-семи функцій приналежності для вихідної змінної виявляється досить для прийнятної якості керування [22].

2.6.5. Реалізація фаззі-керування

Реалізація алгоритму керування в мехатронних системах покладається на фаззі-регулятор. Процедура побудови алгоритму фаззі-регулювання, що виконується поетапно, представлена у вигляді структурної схеми на рис 2.20.

Для простого, але досить розповсюдженого варіанта фаззі-керування із двома вхідними (x_1 і x_2) і однією вихідною (y) змінною масив правил може бути представлений компактно у вигляді таблиці правил (табл. 2.3). Кожна із двох вхідних змінних має п'ять термів (NB , NS , Z , PS , PB), а вихідна змінна, що є керуючим впливом - сім термів (NB , NM , NS , Z , PS , PM , PB) табл. 2.3.

Таблиця 2.3. Масив правил для вхідних та вихідних змінних

x_1 x_2	<i>NB</i>	<i>NS</i>	<i>Z</i>	<i>PS</i>	<i>PB</i>
<i>NB</i>			<i>PM</i>	<i>Z</i>	<i>NS</i>
<i>NS</i>	<i>PB</i>	<i>PM</i>	<i>PS</i>	<i>Z</i>	<i>NS</i>
<i>Z</i>	<i>PM</i>	<i>PS</i>	<i>Z</i>	<i>NS</i>	<i>NM</i>
<i>PS</i>	<i>PS</i>	<i>Z</i>	<i>NS</i>	<i>NM</i>	<i>NB</i>
<i>PB</i>	<i>PS</i>	<i>Z</i>	<i>NM</i>		

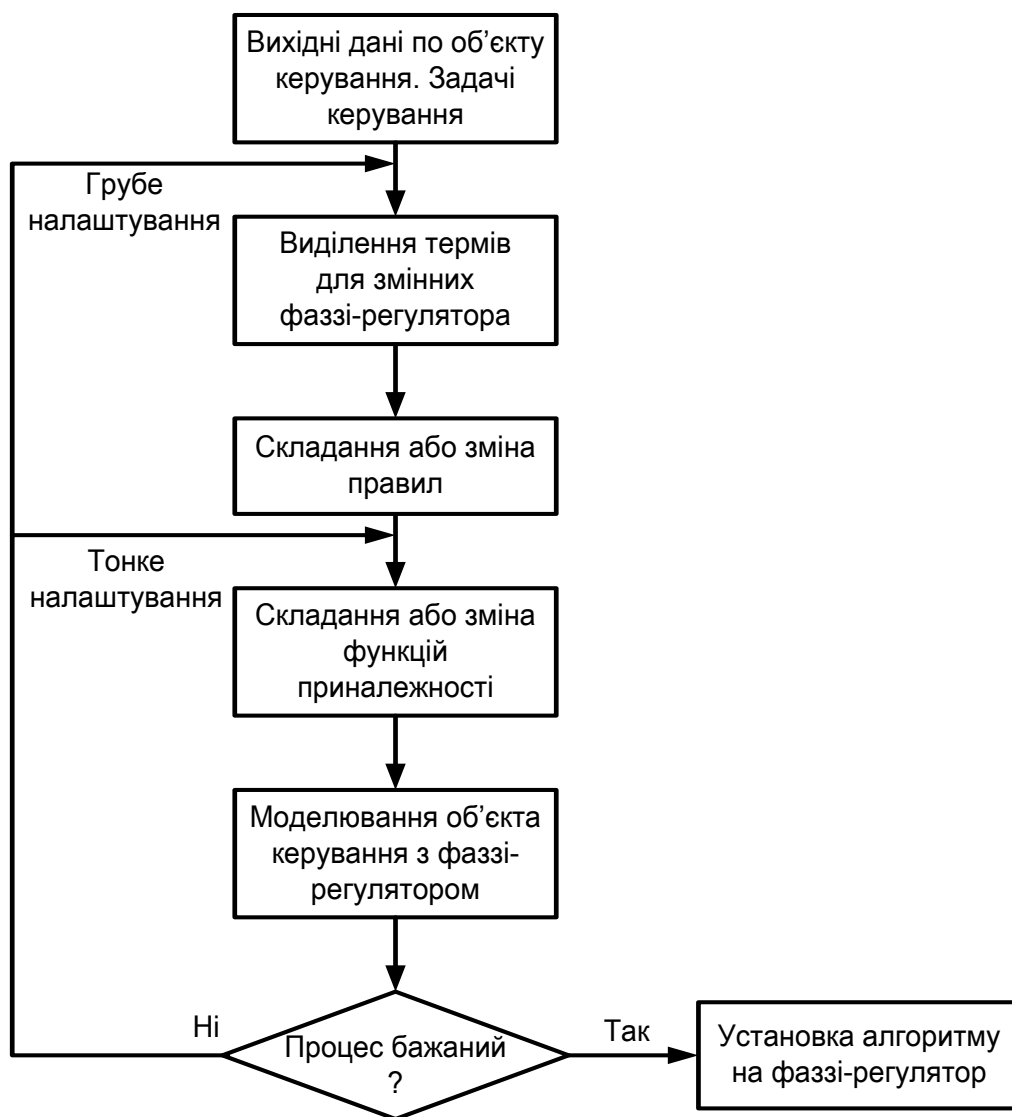


Рис. 2.20. Структурна схема побудови алгоритму фаззі-регулятора

Розглянемо процес перетворення значень вхідних змінних x_1^* і x_2^* у керуючий вплив y^* в системі фаззі-керування, для якої є таблиця правил і функції приналежності (рис. 2.21).

Процедуру формування керуючого впливу, виконувану в три етапи, пояснимо на прикладі дії двох правил (всі правила представлені у таблиці 2.3):

1. ЯКЩО $\tilde{o}_1^* \in A_{12} (NS)$ І $\tilde{o}_2^* \in A_{22} (NS)$, ТО $y_6^* \in B_6 (PM)$
2. ЯКЩО $\tilde{o}_1^* \in A_{11} (NB)$ І $\tilde{o}_2^* \in A_{22} (NS)$, ТО $y_7^* \in B_7 (PB)$.

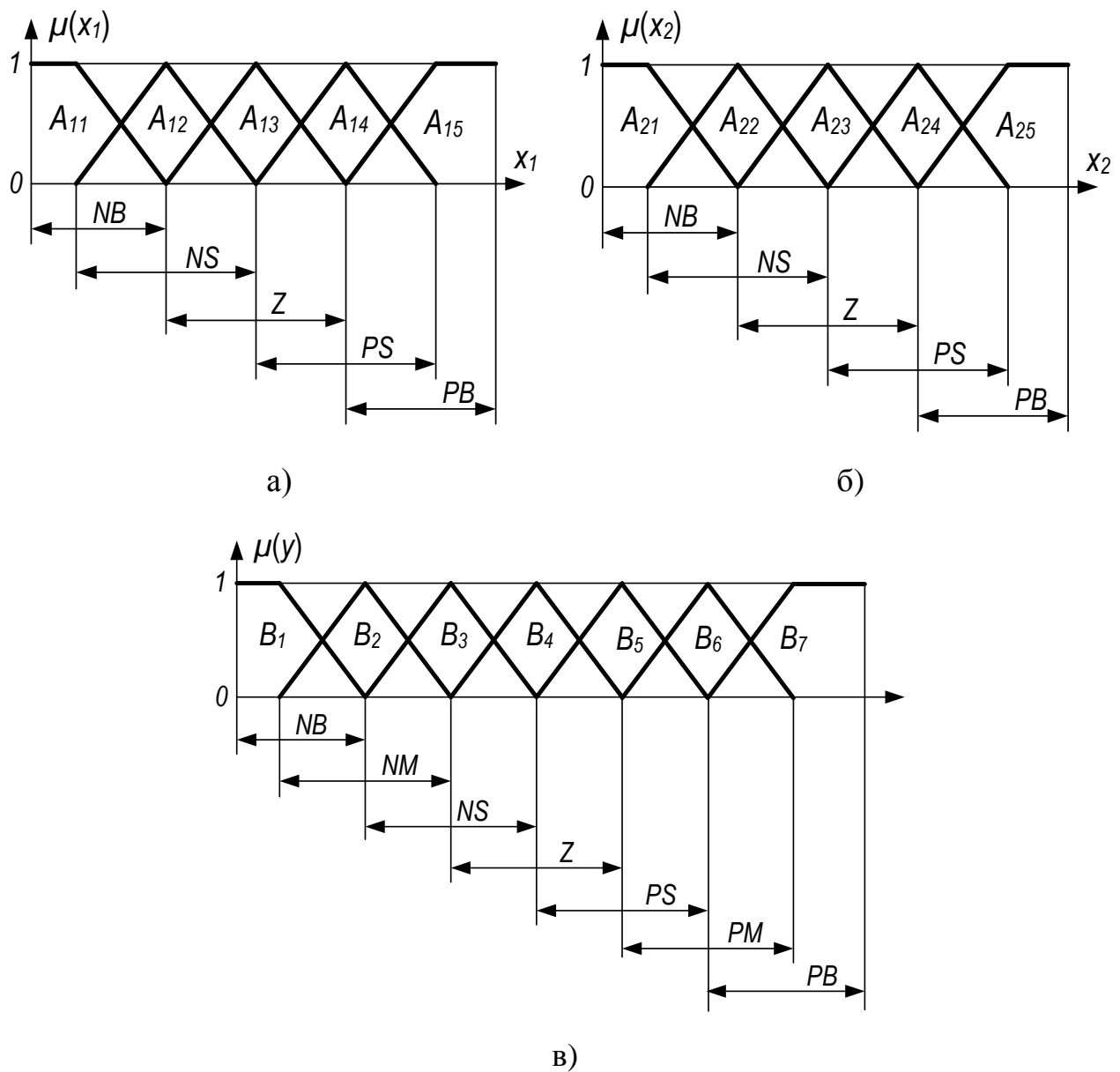


Рис. 2.21. Функції приналежності вхідних x_1 (а), x_2 (б) та вихідної змінної y (в)

На етапі фазифікації вхідні фізичні змінні перетворюються у відповідні терми. Для розповсюдженого на практиці варіанту функції приналежності трикутної форми, коли ліве (a_i) і праве (b_i) граничні значення функції приналежності збігаються відповідно із центром c_{i-1} попередньої функції приналежності та з центром c_{i+1} наступної функції приналежності, тобто коли $a_i=c_{i-1}$; $b_i=c_{i+1}$; $c_i=a_{i-1}=b_{i+1}$ (рис. 2.21), математичний опис функції приналежності має такий вигляд:

$\mu_{\text{в}i.1}(x) = 1$, якщо $x < c_1$;

$\mu_{\text{в}i.3}(x) = \frac{x - c_{i-1}}{c_i - c_{i-1}}$, якщо $c_{i-1} \leq x < c_i$; $i = 2, 3, \dots, m$ (m – число термів

змінної x);

$\mu_{\text{в}i.3}(x) = \frac{x - c_{i+1}}{c_i - c_{i+1}}$, якщо $c_{i+1} \leq x < c_i$; $i = 1, 2, \dots, (m-1)$;

$\mu_{\text{в}i.m}(x) = 1$, якщо $x \geq c_m$.

Для розглянутого прикладу значення x_1^* і x_2^* виміряні в деякий момент часу, визначають два значення функцій приналежності термів A_{12} , A_{22} для правила 1 ($\mu_{12}=0,7$ і $\mu_{22} = 0,4$) і два значення функції приналежності термів A_{11} , A_{22} для правила 2 ($\mu_{11} = 0,3$ та $\mu_{22} = 0,4$).

На етапі логічного висновку вихідним термам змінної y^* присвоюються значення функцій приналежності вхідних змінних із загальною умовою $\mu(y) \leq \mu(x)$ і конкретно для правил з оператором І згідно із принципом мінімуму:

$\mu(y) = \min(\mu_{12}, \mu_{22}) = \mu_{22} = 0,4$ для B_6 (PM) по правилу 1;

$\mu(y) = \min(\mu_{11}, \mu_{22}) = \mu_{11} = 0,3$ для B_7 (PB) по правилу 2.

В результаті вихідні терми $B_6=PM$ і $B_7=PB$ виявляються усіченими мінімальними значеннями вхідних функцій приналежності. Отримані за двома активізованими правилами терми B_6 і B_7 поєднуються за принципом максимуму (оператор АБО) в один загальний терм $B = B_6 \vee B_7$ (рис. 2.22).

У загальному випадку два значення можуть активізувати чотири правила, які об'єднують чотири вихідні терми. Таким чином, на етапі логічного висновку відбувається перетворення вхідних термів у вихідні з передачею від входу до виходу значень функцій приналежності за принципом мінімуму-максимуму, що одержав назву min-max-регулятор Мамдані (по імені його автора). У даному логічному висновку закладений деякий компроміс при прийнятті рішення:

мінімізація вхідних фаззі-величин компенсується певним чином максимізацією вихідної фаззі-величини. Отриманий результуючий терм B визначає множину можливих значень керуючого впливу y .

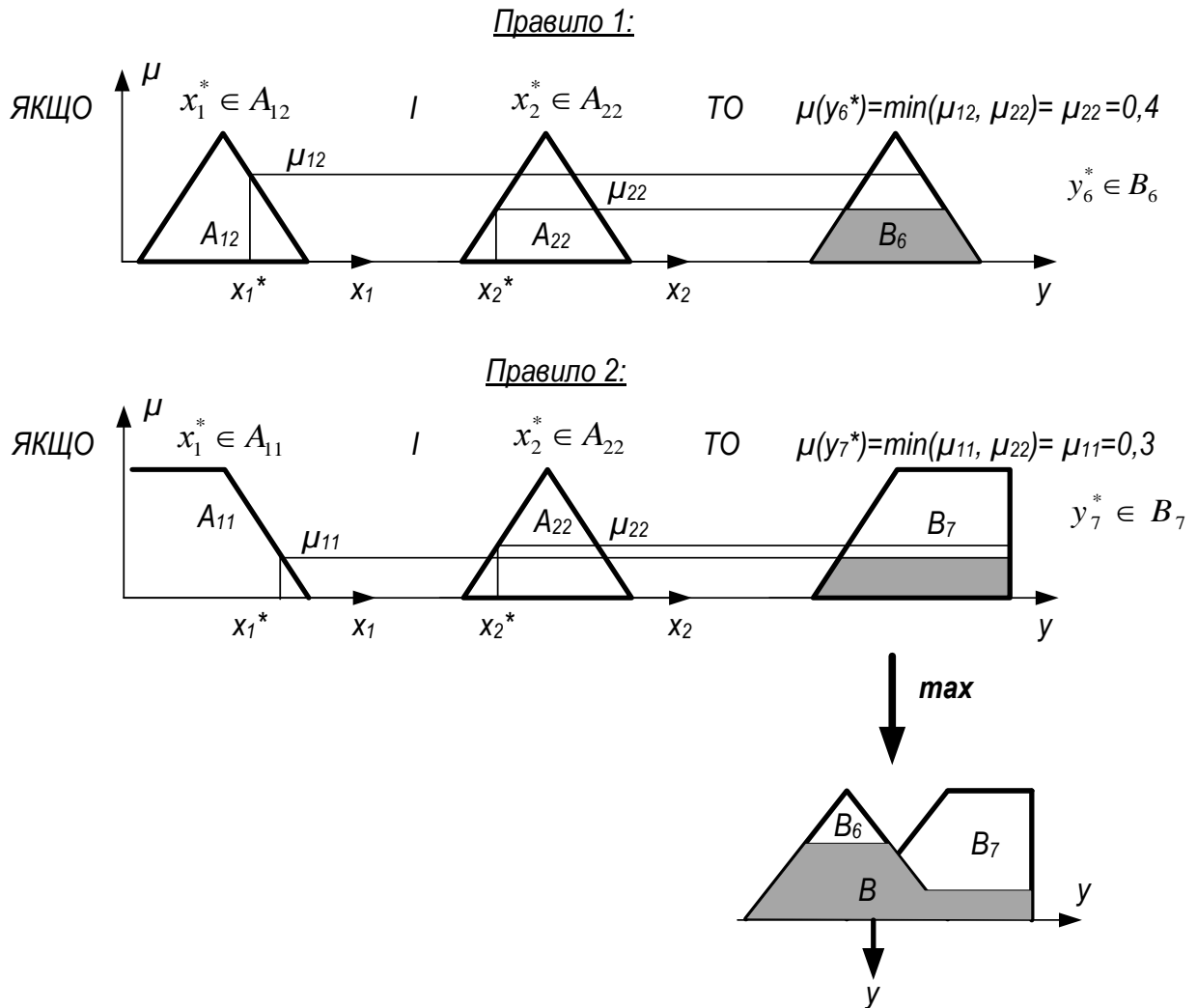


Рис. 2.22. Реалізація двох правил по принципу мінімуму-максимуму

Питання, як вибрати одне конкретне значення y для фізичної вихідної змінної керування, вирішується на третьому етапі перетворення.

На етапі дефаззифікації логічна змінна повинна бути перетворена в певне значення фізичної змінної - керуючого впливу. Нехай активізоване тільки одне правило, наприклад, правило 1. Одному значенню $\mu_6 = 0,4$ усіченого терма B_6 відповідає вся множина значень y від початку терма до його кінця по осі абсцис.

При цьому всі значення в середній частині терма, де $\mu_6=0,4$, рівноцінні по приналежності до терму μ_6 , але різні по величині - у лівій частині вони менші, а в правій більші. Для інтегральної оцінки одним числом усієї множини можливих значень у фаззі-логіці за це число приймають (евристично) абсцису центру ваги площі усіченої частини терма. Для одного правила це виявляється значення, яке відповідає центру терма. Так, для правила 1 $y = \mu_6$.

Практично завжди виявляються активізованими кілька правил, для яких має місце об'єднаний терм B , усічений різними рівнями функцій приналежності. Методом центру ваги усіченої площі знаходиться інтегральне значення у керуючого впливу (рис. 2.22):

$$y = \frac{\int_{y_{\text{лів}}}^{y_{\text{прав}}} y \mu(y) dy}{\int_{y_{\text{лів}}}^{y_{\text{прав}}} \mu(y) dy}. \quad (2.95)$$

Даний метод для розрахунків значення y виявляється складним, вимагає багато місця в пам'яті керуючого пристрою, знижуючи його швидкодію. Тому на практиці, вводячи деякі спрощення в даний метод, використовують його досить нескладну модифікацію у вигляді так званої синглетон-функції:

$$y = \frac{\sum_{i=1}^4 \mu_i c_i}{\sum_{i=1}^4 \mu_i}, \quad (2.96)$$

де μ_i - значення функції приналежності, привласнене i -му вихідному терму; c_i - центр i -го вихідного терма [22].

2.6.6. Особливості фаззі-керування

Лінгвістичні змінні - терми - кількісно оцінюються не окремими числовими значеннями, а числовими множинами, що перекривають одна одну. У цьому сенсі фаззі-керування слід віднести до категорії керування „у великому”.

Фаззі-керування, у принципі, не вимагає точної моделі об'єкта, воно організує наближену стратегію керування, моделюючи спосіб мислення людини. Фаззі-керування дозволяє виразити в простих лінгвістичних термінах будь-який необхідний для визначення керування алгоритм, лінійний або нелінійний, який повинен бути попередньо складений експертом.

Фаззі-контролер, що реалізує керування - це включена в процес керування й працююча в реальному часі експертна система, яка застосовує фаззі-логіку для перетворення якісних логічних змінних - термів. Відзначені особливості дозволяють окреслити коло завдань керування, які можуть ефективно вирішуватись на основі фаззі-логіки для багатьох мехатронних систем.

У загальному випадку можна сказати: фаззі-керування краще використовувати там, де необхідний алгоритм керування простіше сформулювати лінгвістичним шляхом, а в тих областях, де можливі і фаззі- і традиційне керування, перевага віддається тому способу, який дає кращий результат за необхідним показником. Фаззі-керування не підходить для побудови високоточних систем регулювання, однак може успішно використовуватися для зазначених систем як додатковий регулюючий засіб у режимах відпрацьовування великих впливів, коли проявляються нелінійності об'єкта керування й основний регулятор не забезпечує задовільну якість даних режимів. Фаззі-керування здатне надавати нелінійним системам властивість робастності (нечутливості до зміни параметрів системи). Його також доцільно застосовувати як засіб наближеної параметричної адаптації в установках зі змінними параметрами, де точні традиційні системи параметричної адаптації складніші за структурою.

Досить перспективне застосування фаззі-керування в ряді складних і важко описуваних математично технологічних процесах. Для таких об'єктів фаззі-керування, з урахуванням додаткових факторів, що впливають на процес, дозволяє його (процес) поліпшити за різними показниками [22].

2.6.7. Загальна інформація про штучні нейрони

Нервова система біологічних об'єктів складається з нейронів, наприклад, нервова система людини містить у собі від 10^{10} до 10^{12} нейронів 57 модифікацій, розміром від мікрометрів до декількох сантиметрів. Типова форма нейрона представлена на рис. 2.23. Потік електричних сигналів входить у нейрон через його кінцівки (синапси), яких може бути до 1000 на одному дендриті (відгалуженні). Через дендрит інформація надходить у тіло клітини, де відбувається її обробка й оцінка. Результат цієї логічної оцінки (1 або 0) по аксону (стовбуру клітини) передається далі вниз, де інформація розходить по нейронних структурах наступного рівня. Кожний нейрон має зв'язок приблизно з 10^4 інших нейронів. Нервові імпульси передаються як потоки хімічно активних заряджених речовин (іонів).

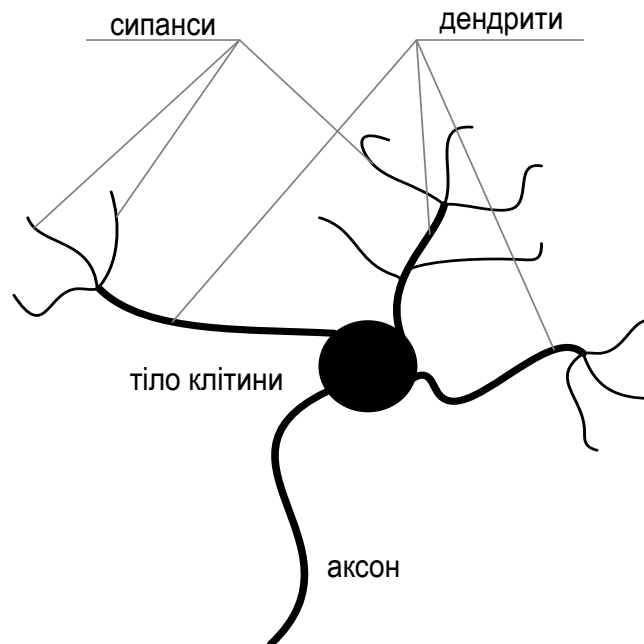


Рис. 2.23. Біологічний нейрон

Штучний нейрон імітує в першому наближенні властивості біологічного нейрона. Він має групу синапсів - вхідних зв'язків, з'єднаних з виходами інших нейронів, а також аксон - вихідний зв'язок даного нейрона, з якого сигнал надходить на синапси інших нейронів. Кожний вхідний сигнал множиться на

відповідний ваговий коефіцієнт і всі добутки додаються, визначаючи рівень активації нейрона.

Загальний вид нейрона, що реалізує ці властивості, наведений на рис. 2.24. Тут вхідні сигнали позначені вектором X . Вагові коефіцієнти, на які множаться відповідні сигнали X , у сукупності позначаються вектором W .

Сумуючий блок (суматор), який відповідає тілу біологічного елемента, складає зважені (помножені на вагові коефіцієнти) сигнали із входів:

$$S = \sum_{i=1}^n x_i w_i, \quad (2.97)$$

де x_i - сигнал із i -того входу; w_i - ваговий коефіцієнт на i -тому вході; n - кількість входів (синапсів) нейрона.

Вихід нейрона є функцією його стану:

$$Y = f(S). \quad (2.98)$$

Нелінійна функція Y називається активаційною й може мати різний вигляд (рис. 2.25).

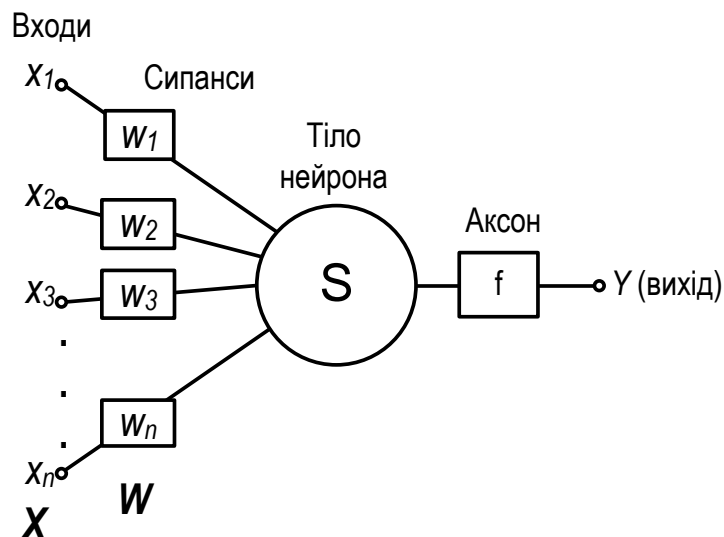


Рис. 2.24. Штучний нейрон

У випадку, коли функція активації однакова для всіх нейронів у мережі, мережу називають однорідною (гомогенною). Якщо ж активаційна функція залежить від одного або декількох параметрів, значення яких змінюються від нейрона до нейрона, то мережу називають неоднорідною (гетерогенною).

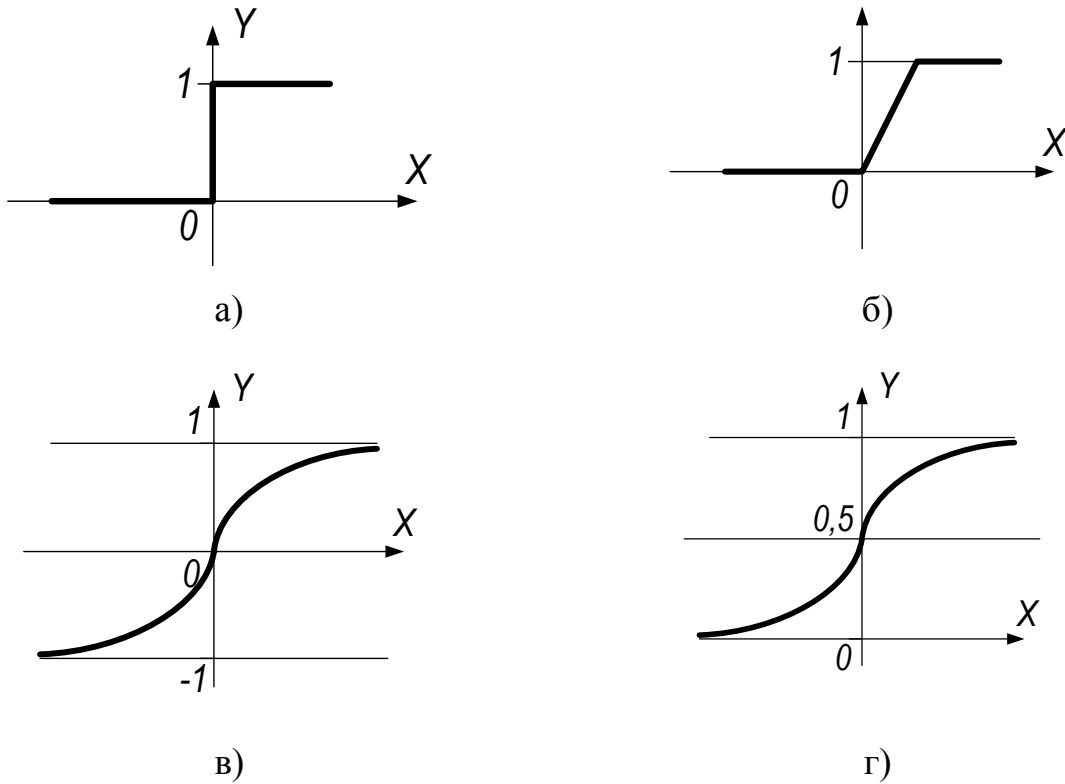


Рис. 2.25. Типи активаційних функцій:

- а – функція одиничного стрибка; б – лінійний поріг (гістерезис);
- в – сигмоїд (гіперболічний тангенс); г – сигмоїд (2.99)

Однією з найпоширеніших активаційних функцій штучних нейронів є нелінійна функція з насиченням, так звана логістична функція або сигмоїд (функція S-подібного виду):

$$f(\tilde{o}) = \frac{1}{1 + \tilde{a}^{-\alpha \tilde{o}}}. \quad (2.99)$$

При зменшенні α сигмоїд стає більш пологим, при граничному переході $\alpha=0$ функція (2.99) вироджується у горизонтальну лінію на рівні 0,5. При збільшенні α сигмоїд наближається по зовнішньому вигляду до функції одиничного стрибка у точці $x=0$. З виразу для сигмоїда очевидно, що вихідне значення сигналу нейрона лежить у діапазоні $[0, 1]$.

Слід зазначити, що сигмоїдна функція диференційована на всій осі абсцис, що широко використовується в багатьох алгоритмах навчання. Крім того, вона має властивість підсилювати слабкі сигнали краще, ніж сильні, і запобігає насиченню від сильних сигналів, тому що вони відповідають областям аргументів, де сигмоїд має пологий нахил. Іншою широко використовуваною активаційною функцією є гіперболічний тангенс. На відміну від логістичної функції гіперболічний тангенс приймає значення різних знаків, що для ряду мереж виявляється бажаною властивістю.

Говорячи про можливу класифікацію нейронних мереж, важливо відзначити існування бінарних і аналогових мереж. Перші з них оперують із двійковими сигналами і вихід кожного нейрона може приймати тільки два значення: логічний нуль („загальмований” стан) і логічна одиниця („збуджений” стан). В аналогових мережах вихідні значення сигналів нейронів здатні приймати неперервні значення.

Ще одна класифікація ділить нейронні мережі на синхронні й асинхронні. У першому випадку в кожний момент часу свій стан міняє лише один нейрон. У другому - стан міняється відразу в цілої групи нейронів, як правило, у всього шару. Для програмних імітаторів нейронних мереж на цифрових ЕОМ, питання пов'язані із синхронізацією вирішуються комп'ютером, на якому реалізується нейронна мережа.

Розглянута проста модель штучного нейрона суттєво спрощує ряд властивостей свого біологічного двійника. Наприклад, у приведеній моделі не

враховуються затримки в часі, які впливають на динаміку системи. Вхідні сигнали відразу ж породжують вихідний сигнал [51].

2.6.8. Архітектура штучних нейронних мереж

Хоча один нейрон і здатний виконувати найпростіші процедури розпізнавання, „сила” нейронних обчислень виникає від з'єднань нейронів у мережі.

Широке коло завдань, яке розв'язується нейронними мережами, не дозволяє в цей час створювати універсальні, потужні мережі, змушуючи розробляти спеціалізовані нейронні мережі, що функціонують за різними алгоритмами.

Вибір структури нейронних мереж здійснюється відповідно до особливостей і складності завдання. Для вирішення деяких окремих типів завдань уже існують оптимальні, на сьогоднішній день, конфігурації нейронних мереж.

Якщо ж завдання не може бути зведено до жодного з відомих типів, розробнику доводиться вирішувати складну проблему синтезу нової конфігурації. При цьому необхідно керуватись декількома основними принципами:

- ▶ можливості мережі зростають зі збільшенням числа нейронів, щільності зв'язків між ними й числа її шарів;
- ▶ введення зворотних зв'язків поряд зі збільшенням можливостей мережі ставить питання про динамічну стійкість мережі;
- ▶ складність алгоритмів функціонування мережі також сприяє посиленню можливостей штучної нейронної мережі.

Питання про необхідні й достатні властивості мережі для розв'язування того або іншого роду завдань являє собою цілий напрямок нейрокомп'ютерної науки. Оскільки, проблема синтезу нейронної мережі сильно залежить від

розв'язуваного завдання, дати загальні докладні рекомендації важко. У більшості випадків оптимальний варіант знаходиться на основі інтуїтивного відбору.

Найпростіша одношарова мережа, що складається із групи нейронів, показана на рис. 2.26.

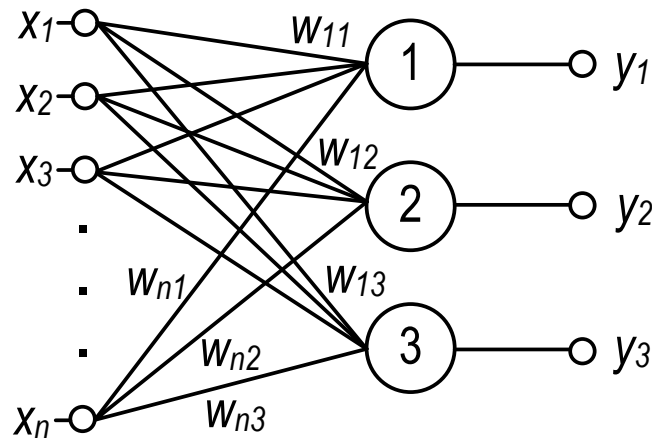


Рис. 2.26. Одношарова нейронна мережа

На n входів надходять сигнали, що проходять по синапсах на три нейрони, які утворюють єдиний шар цієї нейронної мережі і які видають три вихідні сигнали:

$$y_j = f\left(\sum_{i=1}^n x_i w_{ij}\right), \quad (2.100)$$

де $j = 1, 2, \dots, n$. У штучних і природних (біологічних) мережах багато з'єднань можуть бути відсутніми, усі з'єднання на рис. 2.26 приведені з метою показати найзагальніший варіант нейронної мережі.

Очевидно, що всі вагові коефіцієнти синапсів одного шару нейронів можна звести в матрицю \mathbf{W} , у якій кожний елемент w_{ij} задає величину i -того синаптичного зв'язку j -го нейрона. У такий спосіб процес, що відбувається в нейронній мережі, може бути записаний у матричній формі:

$$Y = F(X, W), \quad (2.101)$$

де X і Y - відповідно вхідний і вихідний сигнальні вектори; $F(V)$ - активаційна функція, яка застосовується поелементно до компонентів вектора V .

Вище були розглянуті мережі, які не мали зворотних зв'язків. Відсутність зворотних зв'язків гарантує стійкість мережі. Однак властивість стійкості обмежує можливості мереж прямого поширення в порівнянні з мережами зі зворотними зв'язками. Важливий внесок як у теорію, так і в застосування систем зі зворотними зв'язками зробив Дж. Хопфілд. Тому деякі з конфігурацій нейронних мереж носять його ім'я. Структурна схема мережі Хопфілда наведена на рис. 2.27. Вона складається з єдиного шару нейронів, число яких є одночасно числом входів і виходів мережі.

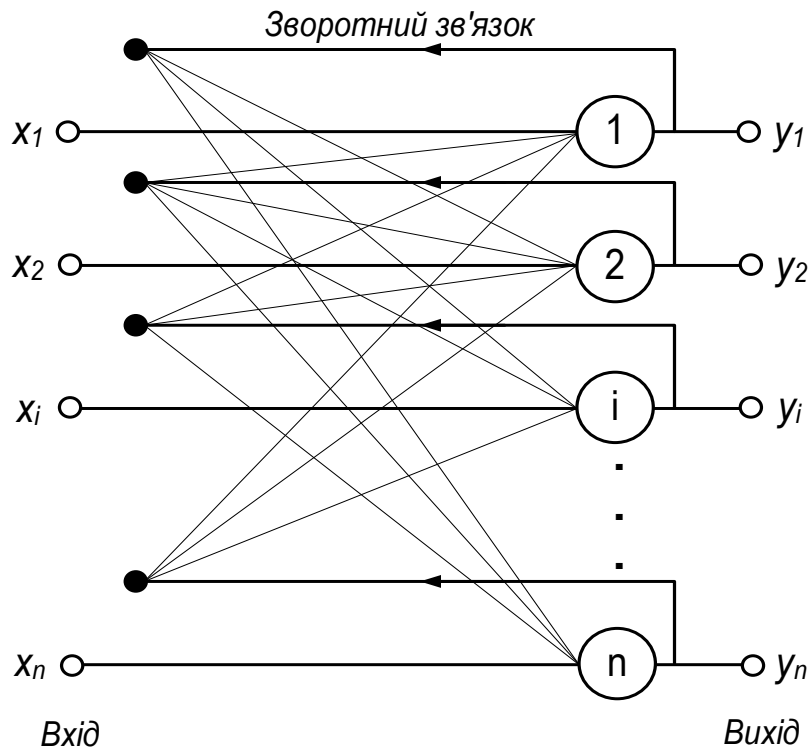


Рис. 2.27. Структурна схема мережі Хопфілда

Кожний нейрон зв'язаний синапсами з усіма іншими нейронами, а також має один вхідний синапс, через який здійснюється введення сигналу. Вихідні сигнали, як звичайно, утворюються на аксонах.

Завдання, розв'язуване даною мережею в якості асоціативної пам'яті, як правило, формулюється в такий спосіб. Відомий деякий набір двійкових сигналів (зображень, звукових оцифровок, інших даних, що описують які-небудь об'єкти або характеристики процесів), які вважаються зразковими. Мережа повинна вміти з довільного неідеального сигналу, поданого на її вхід, виділити („згадати” за частковою інформацією) відповідний зразок (якщо такий є) або „дати висновок” про те, що вхідні дані не відповідають жодному зі зразків [51].

2.6.9. Навчання штучних нейронних мереж

Коли йде розмова про використання нейронних мереж і нейромережних алгоритмів, майже завжди маються на увазі певні процедури їх навчання. Нейронна мережа являє собою адаптивну систему, життєвий цикл якої складається із двох незалежних фаз - навчання й роботи мережі. Навчання вважається закінченим, коли мережа правильно виконує перетворення на тестових прикладах і подальше навчання не викликає значної зміни вагових коефіцієнтів, які під час навчання налаштовуються. Далі мережа виконує перетворення раніше невідомих їй даних на основі сформованої нею в процесі навчання нелінійної моделі процесу. Мережа успішно працює доти, поки суттєво не зміниться реальна модель відображуваного явища (наприклад, у випадку виникнення ситуації, інформація про яку ніколи не пред'являлася мережі при навчанні). Однак, навіть після цього мережа може бути „донавчена” з урахуванням нової інформації, причому при „донавчанні” попередня інформація не губиться, а узагальнюється з отриманою новою. При

„ушкодженні” частини вагових коефіцієнтів нейронної мережі її властивості можуть бути повністю відновлені в процесі „донавчання”.

Від того, наскільки якісно буде виконаний етап навчання нейронної мережі, залежить здатність мережі вирішувати поставлені перед нею завдання під час експлуатації. Теорія навчання розглядає три фундаментальні властивості, пов’язаних з навчанням по прикладах: об’єм, складність зразків і обчислювальна складність. Під об’ємом розуміється, скільки зразків може „запам’ятати” мережа і які функції й границі вирішення завдань можуть бути на ній сформовані. Складність зразків визначає число навчальних прикладів, необхідних для досягнення здатності мережі до узагальнення. Важливою характеристикою є час, який витрачається на навчання. Як правило, час навчання і якість навчання пов’язані зворотною залежністю й вибирати ці параметри необхідно на основі компромісу.

Існують три парадигми навчання: „із вчителем”, „без вчителя” (самонавчання) і змішана.

У свою чергу безліч різних алгоритмів навчання діляться на два великі класи: детерміністських і стохастичних алгоритмів. У першому з них налаштування вагових коефіцієнтів являє собою жорстку послідовність дій, у другому - вона проводиться на основі дій, що описуються деяким випадковим процесом.

Процес навчання „із вчителем” означає, що для кожного вхідного вектора (всіх сигналів, які надходять у нейронну мережу) існує цільовий вектор, що представляє собою необхідний вихід (необхідну величину вихідного сигналу). Разом вони називаються навчальною парою. Звичайно мережа навчається на деякому числі таких навчальних пар.

Коли в мережі тільки один шар, алгоритм її навчання „із вчителем” досить очевидний, тому що правильні вихідні стани нейронів єдиного шару відомі апріорі. Тому налаштування синаптичних зв’язків іде в напрямку, який

мінімізує помилку сигналу на виході мережі (різницю між бажаним сигналом і реально отриманим у процесі навчання).

У багат шарових мережах оптимальні вихідні значення сигналів нейронів усіх шарів, крім останнього, як правило, не відомі, і багат шарову нейронну мережу вже неможливо навчити, керуючись тільки величинами помилок на виходах цієї мережі. Існує кілька варіантів вирішення цієї проблеми:

- ▶ розробка наборів вихідних сигналів, що відповідають вхідним, для кожного шару нейронної мережі, що, звичайно, є дуже трудомісткою операцією, яка не завжди може бути виконана;
- ▶ динамічне налаштування вагових коефіцієнтів синапсів, у ході якої вибираються, як правило, найбільш слабкі зв'язки, які змінюються на малу величину в ту або іншу сторону. Зберігаються ж тільки ті зміни, які спричинили зменшення помилки на виході всієї мережі. Очевидно, що даний метод „проб”, незважаючи на свою гадану простоту, вимагає громіздких рутинних обчислень;
- ▶ поширення сигналів помилки від виходів нейронної мережі до її входів, у напрямку зворотному прямому поширенню сигналів у звичайному режимі роботи. Цей алгоритм навчання нейронної мережі отримав назву процедури зворотного поширення (back-propagation algorithm) і є найбільш широко використовуваним.

Згідно з методом найменших квадратів, цільовою функцією помилки, яку необхідно мінімізувати є величина:

$$E(w) = \frac{1}{2} \sum_{j,p} (y_{j,p}^{(N)} - d_{j,p})^2, \quad (2.102)$$

де $y_{j,p}^{(N)}$ - реальний вихідний стан нейрона у вихідного шару N нейронної мережі при подачі на її входи p -того образу; $d_{j,p}$ – ідеальний (бажаний) вихідний стан цього нейрона.

Операція додавання ведеться по всіх нейронах вихідного шару й по всіх оброблюваних мережею образах. Мінімізація ведеться методом градієнтного спуску, що означає налаштування вагових коефіцієнтів наступним чином:

$$w_{i,j}^{(n)} = -\eta \frac{\partial E}{\partial w_{i,j}}, \quad (2.103)$$

де $w_{i,j}$ - ваговий коефіцієнт синаптичного зв'язку, що з'єднує i -тий нейрон шару $(n-1)$ з j -м нейроном шару n , η - коефіцієнт швидкості навчання, $0 < \eta < 1$. Частина похідна у правій частині виразу (2.103) може бути записана таким чином:

$$\frac{\partial E}{\partial w_{i,j}} = \frac{\partial E}{\partial y_j} \frac{\partial y_j}{\partial s_j} \frac{\partial s_j}{\partial w_{i,j}}. \quad (2.104)$$

У вразі (2.104) під y_j , як і раніше, мається на увазі вихід j -того нейрона, а під s_j - зважена сума його вхідних сигналів, тобто аргумент активаційної функції.

Множник $\frac{\partial y_j}{\partial s_j}$ є похідною цієї функції по її аргументу. Із цього випливає, що похідна активаційної функції повинна бути визначена на всій осі абсцис. У зв'язку із цим функції: одиничного стрибка та інші активаційні функції з неоднорідностями не підходять для розглянутих нейронних мереж. Для них застосовуються такі гладкі функції, як гіперболічний тангенс або класичний сигмоїд з експонентою.

Навчання нейронної мережі „без вчителя” є більш правдоподібною моделлю навчання в біологічній системі. Процес навчання, як і у випадку навчання „із вчителем”, полягає в налаштуванні вагових коефіцієнтів синапсів. Деякі алгоритми передбачають зміну й структури мережі, тобто кількості нейронів і їх зв'язків, але такі перетворення вірніше назвати більш широким терміном - самоорганізацією. Очевидно, що налаштування синапсів може

проводитися тільки на підставі інформації, доступної нейрону, тобто його стану й уже наявних вагових коефіцієнтів. Виходячи із цього міркування й, що більш важливо, за аналогією з відомими принципами самоорганізації нервових клітин, побудовані алгоритми навчання Хебба [51].

2.6.10. Нейрочіпи та нейрокомп'ютери

Швидкий розвиток нейронних мереж та алгоритмів їх функціонування призвів до їх впровадження у промисловість (у тому числі у військову техніку). При цьому спостерігається ріст інтересу до створення засобів апаратної реалізації нейрокомп'ютерної (НК) технології. Насамперед це відноситься до розробки функціонально складних надвеликих інтегральних мікросхем зі спеціалізованою структурою, яка орієнтовано на апаратну реалізацію однієї або декількох найбільш часто використовуваних парадигм нейронних мереж.

Нейрокомп'ютер - це обчислювальна система з архітектурою апаратного й програмного забезпечення, адекватною виконанню алгоритмів, представлених у нейромережному логічному базисі. Системи, створені на стандартній мікропроцесорній елементній базі можуть бути ефективними програмними емуляторами нейромережних алгоритмів. Однак програмна емуляція нейромережних алгоритмів на обчислювальних засобах, реалізованих на елементній базі, що не має відношення до нейромережного логічного базису, скоріше за все, є тимчасовим явищем. Для апаратної реалізації таких алгоритмів ефективніше створювати архітектури, що виконують операції в нейромережному базисі - на основі логічної мережі нейронів при повній відмові від булевих елементів типу І, АБО, НЕ.

Можна виділити наступні труднощі, які виникають при створенні НК:

- висока структурна складність і низька надійність систем;

- практичні труднощі реалізації архітектур навчання, самонавчання, самоорганізації нейронних мереж для безперервно змінюваних вагових коефіцієнтів взаємодії нейронів;
- технологічні складнощі виготовлення надінтегрованих нейронних мереж та НК на нейрочіпах;
- збільшення споживаної потужності й втрата системної швидкодії в підсистемах нейронних мереж у міру збільшення ступеня їх інтеграції.

Вибір ефективного типу нейрокомп'ютера для вирішення конкретного завдання або конкретного класу завдань рекомендується робити в певній послідовності.

На першому етапі в загальноприйнятих термінах формулюється математична постановка завдання (етап формалізації завдання).

На другому етапі здійснюється переформулювання завдання в термінах системної постановки, необхідної для її розв'язку на нейрокомп'ютері. При цьому повинно бути виділене „нейронне ядро” завдання, тобто частина завдання адекватна нейронному базису. Для більш ефективного використання нейрокомп'ютерів це ядро повинне становити основну, як можна більшу, частку операцій в алгоритмі, що представляє собою загальне математичне формулювання завдання.

На третьому етапі для частини алгоритму завдання, що представляє нейронне ядро, формулюються основні характеристики системної постановки завдання: вхідний сигнал, вихідний сигнал, бажаний вихідний сигнал, структура розімкнутої системи, помилка системи, функціонал оптимізації системи, градієнт функціонала оптимізації, алгоритм налаштування системи в процесі розв'язку завдання, замкнена система.

На четвертому етапі для системної постановки конкретного завдання замовника проводиться підрахунок необхідних ресурсів її розв'язку в

структурних термінах нейронних мереж: число нейронів; число зв'язків; число перемикачів зв'язків у процесі розв'язку завдання нейронним ядром алгоритму.

П'ятий етап вимагає проведення моделювання процесу розв'язку завдання на ЕОМ.

На шостому етапі проводиться оцінка продуктивності НК на конкретному завданні з вибором конкретної технології реалізації [51].

2.6.11. Особливості використання штучних нейронних мереж у системах керування

Розробка систем керування завжди відбувається поетапно. У якості таких етапів можна виділити: етап розробки концепції побудови системи керування; етап моделювання системи керування, відповідно до запропонованої концепції побудови; етап аналізу отриманих результатів; етап доробки (модернізації) концепції побудови системи керування. Протягом усіх цих етапів не припиняються теоретичні дослідження, які дозволяють вибирати основні напрямки удосконалення спочатку сформульованої концепції побудови системи керування й поширювати її основні ідеї на ряд суміжних областей.

За аналогічною схемою відбувається розвиток систем керування, що використовують штучні нейронні мережі. Однак слід відмітити, що велика кількість розроблених апаратних і програмних моделей нейромережних систем керування часто випереджають теоретичне розуміння процесів, що відбуваються при цьому, і наявних проблем.

Необхідним етапом розв'язку завдань керування нелінійними динамічними системами є побудова їх адекватних математичних моделей, що базується, як правило, на теоретичному й експериментальному аналізі властивостей цих систем. Теоретичний аналіз процесів, що відбуваються в системі, дозволяє одержати математичний опис у вигляді, наприклад, диференціальних рівнянь. При експериментальному аналізі на основі

спостережень вхідних і вихідних сигналів системи отримують модель. Незважаючи на величезну кількість робіт, різноманіття видів нелінійностей не дозволяє створити єдину теорію ідентифікації нелінійних систем. Найбільш часто застосовується класичний підхід заснований на апроксимації нелінійностей, наприклад, рядами Вольтера, Гаммерштейна, Вінера, поліномами Колмогорова-Габор тощо. Однак область застосування таких моделей обмежена. Крім того, додаткові труднощі одержання адекватного математичного опису обумовлені наявністю в реальних сигналах перешкод.

Однією із класичних моделей систем керування є модель зі зворотним зв'язком та регульованими в реальному масштабі часу коефіцієнтами, наприклад регулятор Астрома із самоналаштуванням. Коефіцієнти такого регулятора підлаштовуються протягом кожного циклу керування відповідно до оцінки параметрів системи. Блок-схема керування зі зворотним зв'язком і регульованими в реальному масштабі часу коефіцієнтами наведена на рис. 2.28.

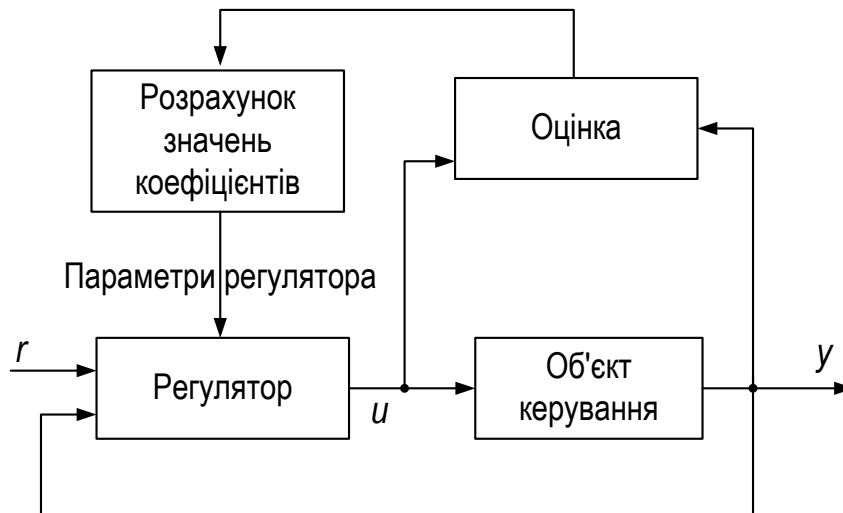


Рис. 2.28. Блок-схема керування зі зворотним зв'язком і регульованими коефіцієнтами

На рис. 2.28: r – уставка, u – керування, y – узагальнена координата руху мехатронної системи.

Іншою добре відомою моделлю системи керування є модель Ляпунова. Системи адаптивного керування, що використовують еталонну модель Ляпунова проектують таким чином, щоб вихідний сигнал керованої моделі зрештою відповідав вихідному сигналу попередньо визначеної моделі, яка має бажані характеристики. Така система повинна бути асимптотично стійкою, тобто керована система відслідковує еталонну модель із нульовою помилкою. Більше того, перехідні процеси на етапі адаптивного керування або навчального керування мають гарантовані межі. Блок-схема адаптивного керування з еталонною моделлю представлена на рис. 2.29.

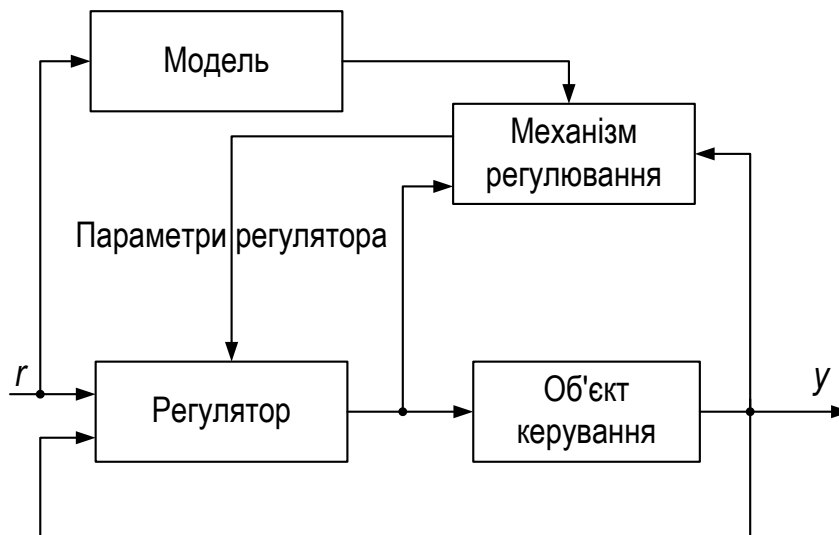


Рис. 2.29. Блок-схема адаптивного керування з еталонною моделлю

Системи керування, що використовують штучні нейронні мережі є однією з можливих альтернатив класичним методам керування. Можливість використання нейронних мереж для розв'язування завдань керування багато в чому ґрунтується на тому, що нейронна мережа, яка складається із двох шарів, може апроксимувати будь-яку функцію дійсних чисел із заданим ступенем точності. Доказ цього положення, оснований на відомій теоремі Вейерштрасса.

Одним з перших методів, які використовувались для побудови нейромережних систем керування був метод, заснований на „копіюванні”

існуючого контролера. Застосувавши цей метод в 1964 році Уїдроу назвав його методом побудови експертної системи за рахунок одержання знань від уже існуючого експерта. Архітектура такої системи керування представлена на рис. 2.30.



Рис. 2.30. Нейромережна система керування, заснована на „копіюванні” існуючого контролера

Дивлячись на рис. 2.30 можна засумніватися в корисності цього методу. Навіщо використовувати ще один керуючий контролер (у вигляді нейронної мережі), якщо один вже існує? Однак, по-перше, існуючий контролер може бути незручний у використанні (наприклад, у ролі такого контролера може виступати людина), а по-друге, для виконання ефективного керування нейронна мережа може використовувати відмінну, від існуючого контролера, інформацію про стан об'єкта керування, яку, наприклад, легше виміряти, формалізувати тощо.

У цей час достатньо добре розроблений і широко використовується цілий ряд інших можливих архітектур побудови нейромережних систем керування. Їх призначення – виконання адекватного керування динамікою стану об'єкта керування від початкового стану до бажаного кінцевого стану, причому зміна станів повинна відбуватися по оптимальній траєкторії. Організація контролю над станом об'єкта керування й реалізація нейромережного контролера в

значній мірі залежать від обраного алгоритму навчання й використовуваної структури керування. Найбільш широко використовуваними є схеми прямого (безпосереднього) й непрямого керування. При цьому найчастіше в якості алгоритму навчання використовується алгоритм зворотного поширення помилки.

У схемі непрямого керування параметри об'єкта керування визначаються в кожний момент часу й потім використовуються для оцінки параметрів контролера (рис. 2.31). Таким чином, має місце явно виражений процес ідентифікації об'єкта керування.

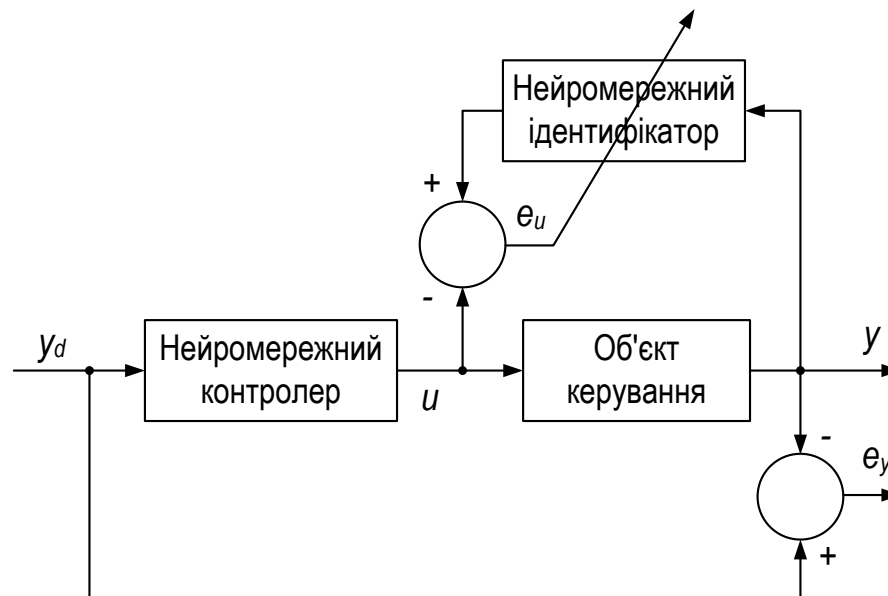


Рис. 2.31. Схема непрямого керування

Недоліком такої схеми є те, що ідентифікація й керування ґрунтуються винятково на помилці керування e_u , і, отже, мінімізацію помилки відпрацювання узагальненої координати e_y на виході системи гарантувати не можна.

У схемі прямого керування параметри нейромережного контролера регулюються таким чином, щоб зменшити безпосередньо помилку виходу e_y (рис. 2.32).

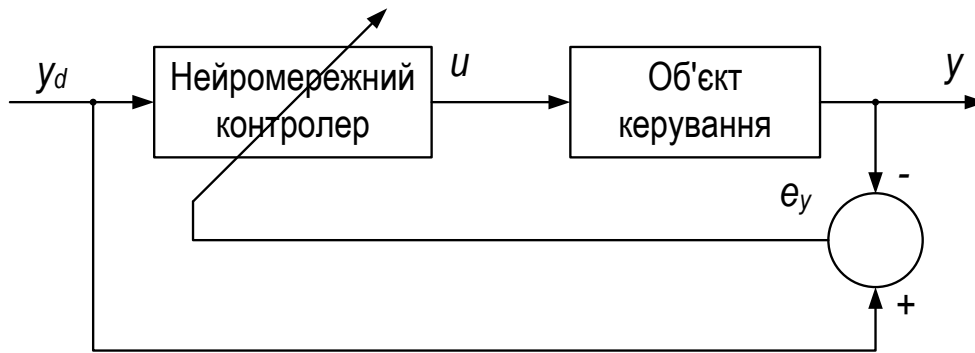


Рис. 2.32. Схема прямого керування

У якості цільової функції, яка повинна бути мінімізована контролером, використовується середньоквадратична помилка на виході об'єкта керування

$$\dot{a}_o = \frac{1}{2}(y_d - y)^2. \quad (2.105)$$

При використанні нейронних мереж відсутні обмеження на лінійність системи, цей метод ефективний в умовах шумів і після закінчення навчання забезпечує керування в реальному масштабі часу. Нейромережні системи керування більш гнучко налаштовуються на реальні умови, утворюючи моделі повністю адекватні поставленому завданню, що не містять обмежень, пов'язаних з побудовою формальних систем. Крім того, нейромережні системи керування не тільки реалізують стандартні адаптивні методи керування, але й пропонують свої алгоритмічні підходи до ряду завдань, розв'язок яких викликає складнощі внаслідок неформалізованості.

Контролери на основі нейронних мереж ефективні у випадках, коли створення адекватної аналітичної моделі виконавчої системи й синтез на її основі регуляторів у край складний. Така ситуація може бути обумовлена цілим рядом факторів, серед яких найпоширенішими є:

- ▶ наявність заздалегідь невизначених зовнішніх впливів (наприклад, при роботі машини в екстремальних середовищах);
- ▶ непостійність параметрів і структури самої мехатронної системи;
- ▶ істотні збурюючі внутрішні впливи (наприклад, дія сил сухого й в'язкого тертя в механічних пристроях);
- ▶ складні фізичні (зокрема, динамічні) взаємозв'язки між елементами системи;
- ▶ технічні й методичні проблеми з постановкою й проведенням експериментальних досліджень на реальних об'єктах для ідентифікації параметрів математичної моделі з необхідною точністю.

Майбутнє інтелектуального керування лежить у комбінації традиційного керування з потенційними можливостями й перспективами використання систем, заснованих на використанні штучних нейронних мереж [51].

2.7. Питання та завдання для самоперевірки

1. Поясніть у чому полягає ієрархія мехатронних систем керування?
2. Які етапи у проектуванні систем керування мехатронними об'єктами можна виділити?
3. Поясніть поняття „стійкість” системи та приведіть приклади стійких та нестійких динамічних систем.
4. Вкажіть як можна класифікувати керування залежно від його цілей.
5. Що таке адаптивні системи керування та які контрольовані зміни можуть в них відбуватись?
6. Поясніть сутність показників якості автоматичного регулювання.
7. Які закони регулювання вам відомі?
8. Прокоментуйте основні особливості реальних ПІД-регуляторів?
9. Розкрийте суть методів налаштування ПІД-регуляторів.
10. Які методи оптимального керування вам відомі?
11. Приведіть математичну постановку задачі оптимальної швидкодії системи яка описується диференціальним рівнянням $m\ddot{x} = F$.
12. Запишіть рівняння Ейлера для функціонала $\int_0^{\dot{\circ}} \left(x^2 + \frac{\dot{x}^2}{2} \right) dt$.
13. Чому принцип максимуму Понтрягіна є більш загальним методом аніж варіаційне числення?
14. Прокоментуйте принцип оптимальності Беллмана.
15. Які основні елементи повинна містити в собі система штучного інтелекту.
16. Яке місце займають нечіткі систем у сфері методів керування?
17. Поясніть структуру системи фаззі-керування.
18. Дайте тлумачення поняттям „терм” та „функція приналежності”.
19. Які типи активаційних функцій штучних нейронів вам відомі?
20. Вкажіть які структури штучних нейронних мереж вам відомі.
21. Що таке нейрочіп та нейрокомп'ютер?

3. СПОСОБИ ОРГАНІЗАЦІЇ ВЗАЄМОДІЇ МІЖ ЕЛЕМЕНТАМИ МЕХАТРОНОЇ СИСТЕМИ

3.1. Основні функції, будова та параметри системи комп'ютерного керування

На систему комп'ютерного керування мехатронного об'єкта покладають наступні основні функції:

- приймання даних, одержуваних за допомогою інформаційних мехатронних модулів (наприклад, даних про стан мехатронного об'єкта й навколишнього середовища);
- обробка цих даних згідно алгоритмів керування мехатронного об'єкта;
- пересилання команд керування, одержаних у результаті реалізації алгоритму керування, на виконавчі мехатронні модулі.

Для реалізації цих функцій до складу система комп'ютерного керування повинні входити:

- обчислювальний пристрій, що служить для реалізації алгоритму розв'язку завдання керування;
- інтерфейсний пристрій, що служить для обміну даними й командами керування з інформаційними та виконавчими системами.

Залежно від способів організації обчислювального й інтерфейсного пристрою відповідно змінюється й загальна організація системи комп'ютерного керування. Розглянемо й порівняємо різні способи організації системи комп'ютерного керування мехатронного об'єкта.

Для такого порівняння, необхідно визначити критерії, за якими воно повинно здійснюватися. До системи комп'ютерного керування мехатронного об'єкта може пред'являтися ряд різних вимог, наприклад:

- вимоги до швидкодії;
- вимоги до надійності;
- вимоги (обмеження) за вартістю;
- вимоги (обмеження) щодо споживаної потужності;

- вимоги (обмеження) щодо масогабаритних характеристик;
- вимоги щодо стійкості впливу зовнішніх факторів;
- вимоги, що обмежують негативний вплив на зовнішнє середовище.

На організацію й структуру системи комп'ютерного керування, власне кажучи, істотний вплив виявляють тільки перші три з перерахованих вимог, у той час як інші вимоги виявляються або залежними від перших трьох (наприклад, вимоги до маси й габаритів системи комп'ютерного керування), або задовольняються за рахунок конструктивних рішень (наприклад, вимоги щодо стійкості впливу зовнішніх факторів). Тому подальше дослідження й порівняння різних способів організації системи комп'ютерного керування мехатронного об'єкта будемо здійснювати саме на основі вимог щодо швидкодії, надійності й вартості. Відмітимо, що виділені нами вимоги є найпоширенішими при проектуванні складних мехатронних об'єктів.

Для однозначності подальших міркувань необхідно визначити, що слід розуміти під швидкодією, надійністю й вартістю системи комп'ютерного керування.

Під швидкодією системи комп'ютерного керування будемо розуміти величину, обернено пропорційну часу розв'язування задач керування мехатронним об'єктом. Чим менший цей час, тим більше буде швидкодія системи комп'ютерного керування. Очевидно, що час розв'язування завдання керування системи комп'ютерного керування складається із часу приймання вихідних даних від інформаційної системи, часу реалізації алгоритму керування й часу передачі команд керування на виконавчу систему. Тому швидкодія системи комп'ютерного керування в цілому залежить як від швидкодії обчислювальної частини, так і від швидкодії інтерфейсної частини.

Надійність - це властивість системи зберігати в часі здатність виконання необхідних функцій.

У нашому випадку під цією функцією слід розуміти здатність системи комп'ютерного керування вирішувати задачі керування за час, що не

перевищує заданий. Очевидно, що ця здатність і буде залежати від здатності інтерфейсної частини передавати дані між інформаційною й виконавчою системами мехатронного об'єкта й обчислювальною частиною системи комп'ютерного керування, а також від здатності обчислювальної частини пристрою системи комп'ютерного керування реалізувати алгоритм керування за необхідний час. Іншими словами, надійність системи комп'ютерного керування також „складається” з надійності обчислювального пристрою та надійності інтерфейсної частини.

Для кількісної оцінки надійності використовують різні критерії, такі як імовірність безвідмовної роботи, середній наробіток на відмову та інші. З перерахованих показників для характеристики надійності системи комп'ютерного керування, що працюють у критичних умовах, найбільш зручно застосовувати показник імовірності безвідмовної роботи.

Імовірність безвідмовної роботи - це ймовірність того, що в межах заданого часу (тривалості) роботи системи відмова не відбудеться. У нашому випадку під відмовою слід розуміти неможливість системи комп'ютерного керування виконувати необхідну функцію. Будемо вважати, що рівень надійності системи комп'ютерного керування достатній, якщо ймовірність її безвідмовної роботи вище заданої.

Під вартістю системи комп'ютерного керування будемо розуміти вартість апаратних засобів, необхідних для її реалізації. Очевидно, що вартість системи комп'ютерного керування складається з вартості апаратних засобів обчислювальної та інтерфейсної частин.

Оцінимо різні способи організації системи комп'ютерного керування мехатронного об'єкта з погляду виділених критеріїв. Розглянемо спочатку різні способи організації частин системи комп'ютерного керування окремо, а потім зведемо отримані оцінки й загальну оцінку системи комп'ютерного керування [52].

3.2. Способи організації обчислювального пристрою системи комп'ютерного керування

Можна виділити три способи організації обчислювального пристрою системи комп'ютерного керування (рис. 3.1).



Рис. 3.1. Способи організації обчислювального пристрою системи комп'ютерного керування

Однопроцесорна централізована організація обчислювального пристрою системи комп'ютерного керування означає наявність одного процесорного вузла, на який покладається функція реалізації алгоритму керування мехатронного об'єкта (рис. 3.2).

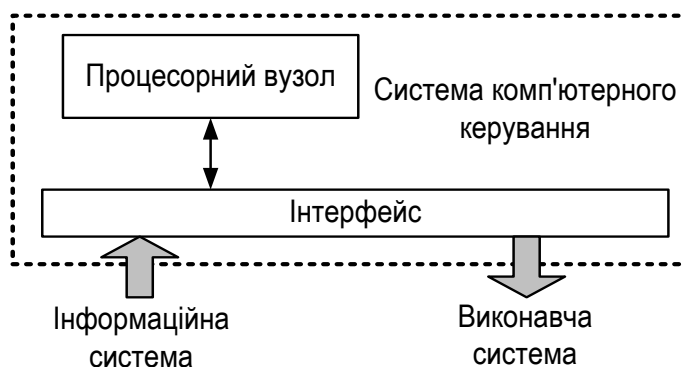


Рис. 3.2. Структура системи комп'ютерного керування з однопроцесорною організацією обчислювального пристрою

До переваг такої системи комп'ютерного керування можна віднести простоту її організації й відносно низьку вартість. Однак, з іншого боку, така організація системи комп'ютерного керування має й ряд істотних недоліків, а саме: 1) низька надійність, оскільки відмова процесорного вузла приводить до відмови всієї системи комп'ютерного керування (тобто до

нездатності системи комп'ютерного керування виконувати задану функцію); 2) низька швидкодія, оскільки реалізація алгоритму керування в цьому випадку буде здійснюватися послідовним способом.

Перераховані переваги й недоліки однопроцесорної організації системи комп'ютерного керування мехатронного об'єкту визначають і основну сферу їх застосування, а саме, у нескладних мехатронних об'єктах, які працюють в некритичних умовах експлуатації, наприклад: побутові мехатронні об'єкти, верстати зі ЧПК (рис. 3.3), телескопи (рис. 3.4) тощо.

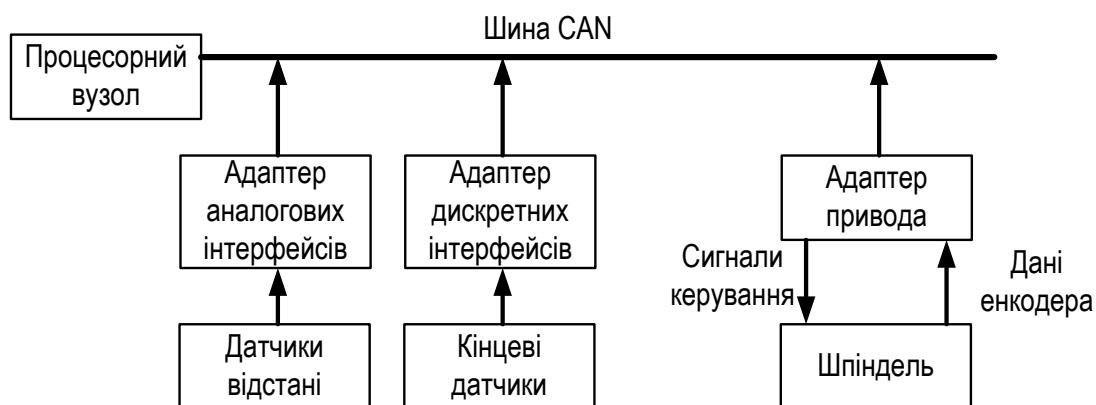


Рис. 3.3. Структура сучасного верстата зі ЧПК

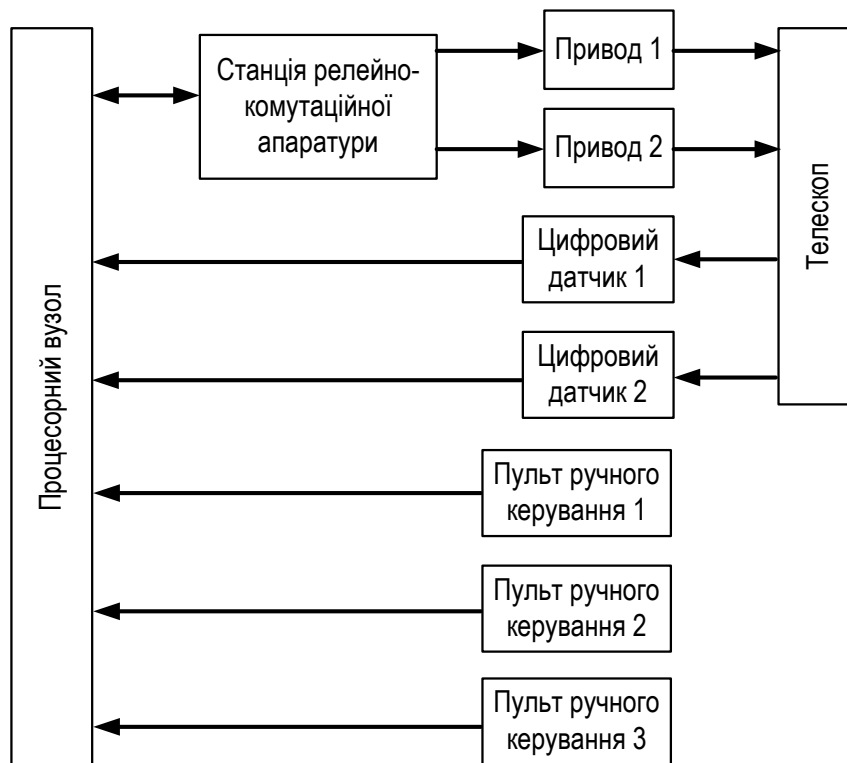


Рис. 3.4. Структура телескопа

У випадку керування складним мехатронним об'єктом швидкодія однопроцесорної системи комп'ютерного керування, як правило, виявляється недостатньою для керування в реальному часі. У цьому випадку використовують розподілений ієрархічний спосіб організації системи комп'ютерного керування. Узагальнена структура системи комп'ютерного керування, що використовує такий спосіб організації обчислювального пристрою, наведена на рис. 3.5.

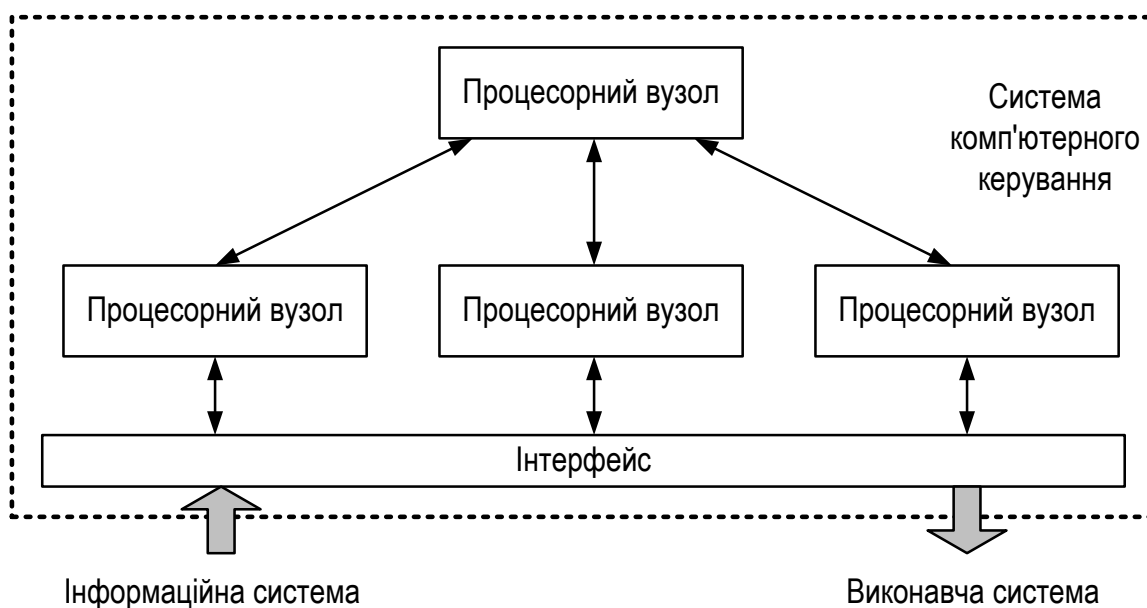


Рис. 3.5. Структура системи комп'ютерного керування з розподіленою ієрархічною організацією обчислювального пристрою

Обчислювальний пристрій системи комп'ютерного керування будується у вигляді багаторівневої ієрархічної мережі процесорних вузлів, у якій кожному процесорному вузлу верхнього рівня підкоряється кілька процесорних вузлів нижнього рівня (на рис. 3.5 показано два рівні).

Очевидно, що в порівнянні з однопроцесорною організацією така організація обчислювального пристрою системи комп'ютерного керування має ряд переваг:

- по-перше, за рахунок „розпаралелювання” процедур реалізації алгоритму керування між окремими процесорними вузлами забезпечується більш висока швидкодія;
- по-друге, підвищується надійність обчислювального пристрою, оскільки відмова деяких процесорних вузлів нижнього рівня може не приводити до катастрофічних наслідків для всього мехатронного об’єкта в цілому.

Однак недоліки в такій організації обчислювального пристрою системи комп’ютерного керування теж є:

- по-перше, суттєво (не менше ніж у N разів, де N - число процесорних вузлів в системі) збільшується вартість ієрархічного розподіленого пристрою в порівнянні з однопроцесорною організацією обчислювального пристрою;
- по-друге, так само в системі залишається „вузьке місце”, а саме, процесорний вузол верхнього рівня, відповідальний за організацію спільної роботи всіх процесорних вузлів нижнього рівня. Зрозуміло, що відмова процесорного вузла верхнього рівня буде приводити до виходу з ладу всього мехатронного об’єкта в цілому;
- по-третє, у даній структурі взаємодія процесорних вузлів нижнього рівня один з одним може відбуватися тільки через процесорний вузол верхнього рівня, що, у свою чергу, може приводити до значних часових затримок при обміні даними між ними й, як наслідок, до зниження швидкодії системи комп’ютерного керування в цілому.

У якості прикладів мехатронних об’єктів, що мають розподілену ієрархічну організацію системи комп’ютерного керування можна привести прискорювач частот (рис. 3.6), космічну станцію (рис. 3.7).

Перерахованих недоліків дозволяє уникнути спосіб розподіленої децентралізованої організації обчислювального пристрою системи комп’ютерного керування.

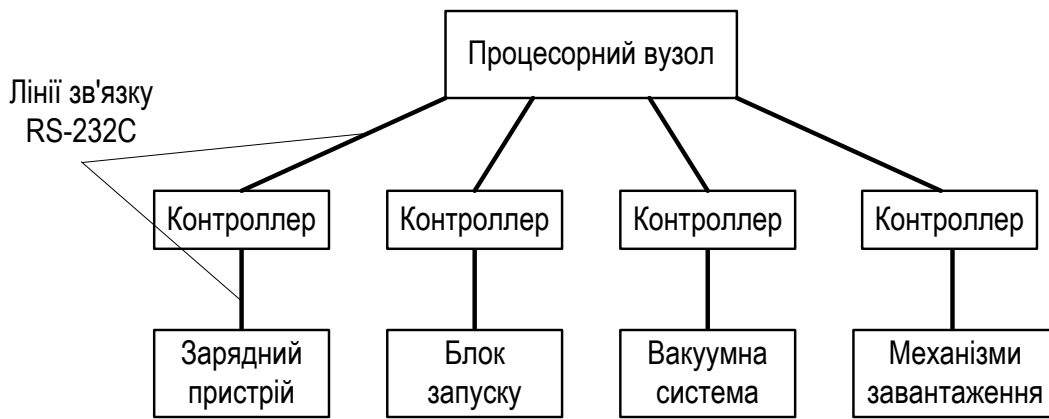


Рис. 3.6. Структура потужнострумового імпульсного прискорювача електронів ГЕЗА

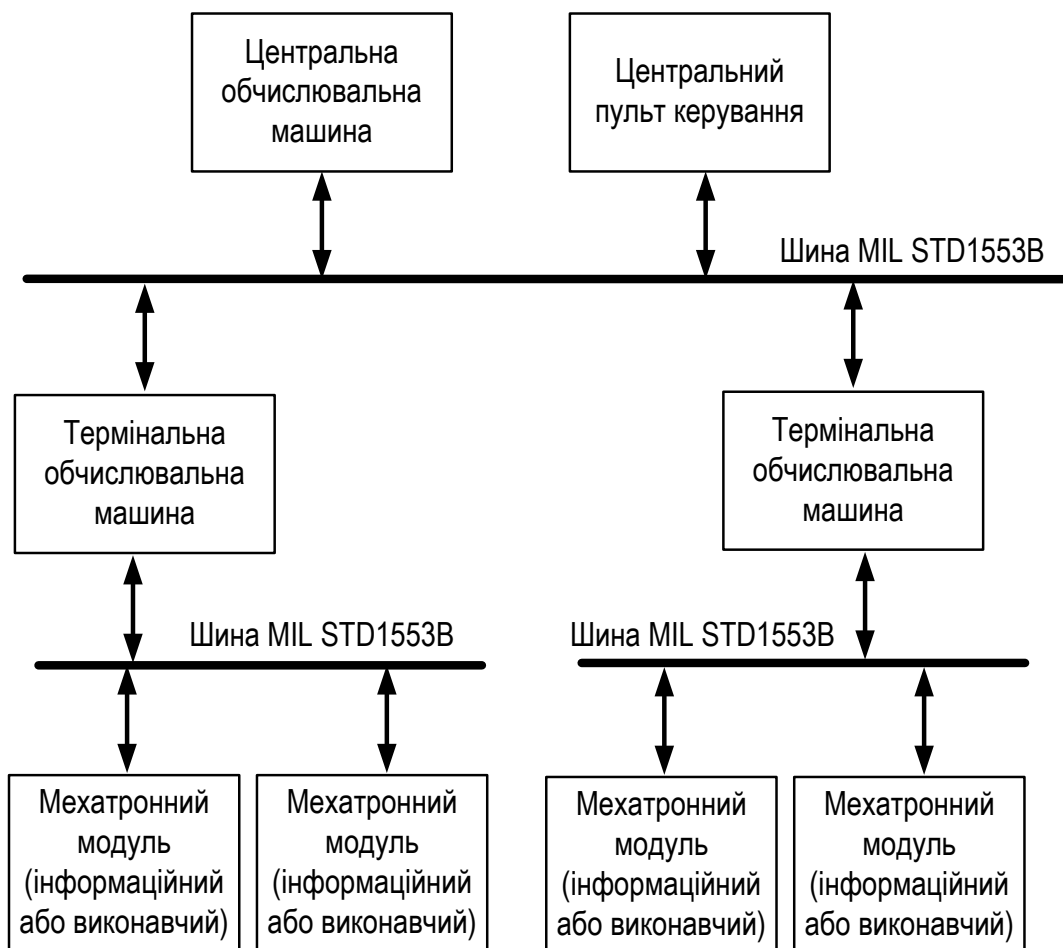


Рис. 3.7. Структура системи комп'ютерного керування космічної станції

На відміну від розподіленої ієрархічної організації системи комп'ютерного керування при розподіленій децентралізованій організації

системи комп'ютерного керування (рис. 3.8) відсутня яка-небудь підпорядкованість одних процесорних вузлів перед іншими. У принципі, усі процесорні вузли, що входять в обчислювальний пристрій системи комп'ютерного керування, можуть бути рівні з погляду функціональних можливостей. Така організація системи комп'ютерного керування мехатронного об'єкта, яка в літературі називається колективом обчислювачів, відкриває перспективи досягнення цілого ряду переваг:

- по-перше, система буде мати дуже високу надійність. Дійсно, оскільки всі процесорні вузли рівні, то вихід з ладу кожного з них не буде приводити до катастрофічних наслідків, а функції процесорного вузла, який вийшов з ладу, можуть бути перерозподілені між працездатними процесорними вузлами;
- по-друге, забезпечується можливість прямого обміну даними між будь-якими парами процесорних вузлів, а не через процесорний вузол верхнього рівня, як при ієрархічній організації, що суттєво знижує тимчасові втрати й, відповідно, збільшує швидкодію обчислювального пристрою.

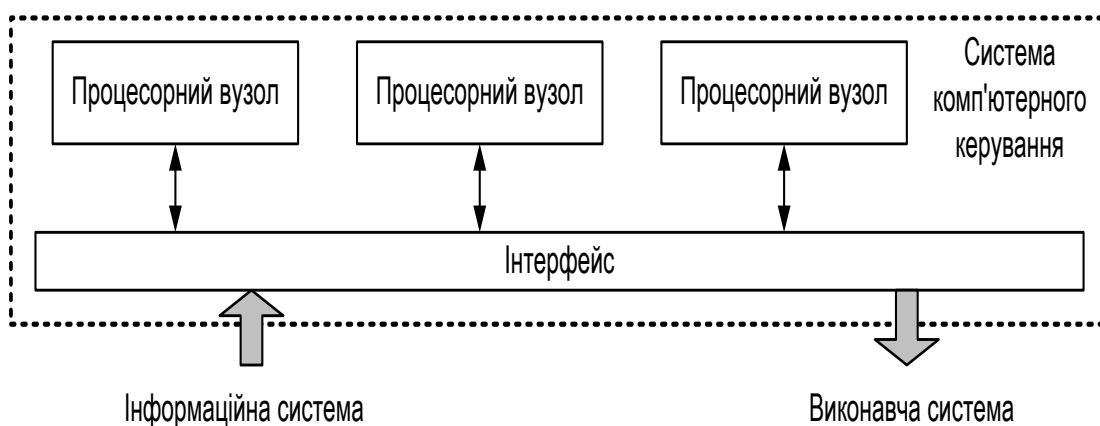


Рис. 3.8. Структура системи комп'ютерного керування з розподіленою децентралізованою організацією обчислювального пристрою

У той же час, незважаючи на явно виражені потенційні переваги розподіленої децентралізованої організації системи комп'ютерного

керування, практично відсутні приклади її застосування в реальних мехатронних об'єктах. Це викликане, очевидно, складнощами організації обчислювального процесу при розв'язку завдань керування мехатронним об'єктом в такому колективі рівноправних процесорних вузлів.

У таблиці 3.1 зведені якісні оцінки розглянутих способів організації обчислювального пристрою системи комп'ютерного керування за прийнятим вище критеріям [52].

Таблиця 3.1. Способи організації взаємодії та їх характеристики

Спосіб організації	Швидкодія	Надійність	Вартість
Однопроцесорний (централізований)	Низька	Низька	Низька
Розподілений ієрархічний	Середня	Середня	Висока
Розподілений децентралізований	Висока	Висока	Висока

3.3. Варіанти організації інтерфейсної частини системи комп'ютерного керування

Розглянемо тепер варіанти організації інтерфейсної частини системи комп'ютерного керування з погляду критеріїв швидкодії, надійності й вартості. При цьому під швидкістю інтерфейсу будемо розуміти величину, обернено пропорційну часу передачі даних між виконавчою й інформаційною підсистемами мехатронного об'єкта й обчислювальною частиною системи комп'ютерного керування в процесі розв'язування задач керування. Під надійністю інтерфейсу будемо розуміти здатність передачі безпомилкових даних у процесі розв'язування задач керування. Нарешті, під вартістю інтерфейсу будемо розуміти вартість апаратних засобів, необхідних для його реалізації.

Можна виділити три типи інтерфейсів, які можуть використовуватися для обміну даними між обчислювальним пристроєм системи комп'ютерного керування й іншими підсистемами мехатронного об'єкта (рис. 3.9).



Рис. 3.9. Типи інтерфейсів системи комп'ютерного керування мехатронного об'єкта

Узагальнена структура мехатронного об'єкта при використанні аналогових інтерфейсів наведена на рис. 3.10.

Аналоговий інтерфейс означає використання в складі мехатронного об'єкта інформаційних і виконавчих модулів першого покоління, обмін даними з якими здійснюється у формі електричного сигналу. Для перетворення аналогових даних, що надходять із інформаційної підсистеми, у цифрові, доступні для обробки в обчислювальному пристрої, до складу системи комп'ютерного керування необхідно ввести ряд аналогово-цифрових перетворювачів (АЦП). Аналогічно, для перетворення цифрових команд керування, сформованих обчислювальним пристроєм, в аналогову форму, доступну для приводів виконавчих мехатронних модулів, використовується ряд цифро-аналогових перетворювачів, які в цьому випадку також повинні входити до складу системи комп'ютерного керування.

До недоліків аналогового інтерфейсу слід віднести його низьку надійність, викликану слабкою перешкодозахищеністю, і обмеження на довжину ліній зв'язку, а також високу вартість необхідної кабельної мережі, оскільки кожне джерело й кожний приймач даних повинні бути з'єднані окремою лінією зв'язку.

У той же час, аналогові інтерфейси відносно дешеві, що обумовлює їх досить широке застосування в побутових мехатронних приладах, у яких

немає довгих ліній зв'язку для передачі аналогових сигналів і до яких не пред'являються підвищені вимоги по надійності.

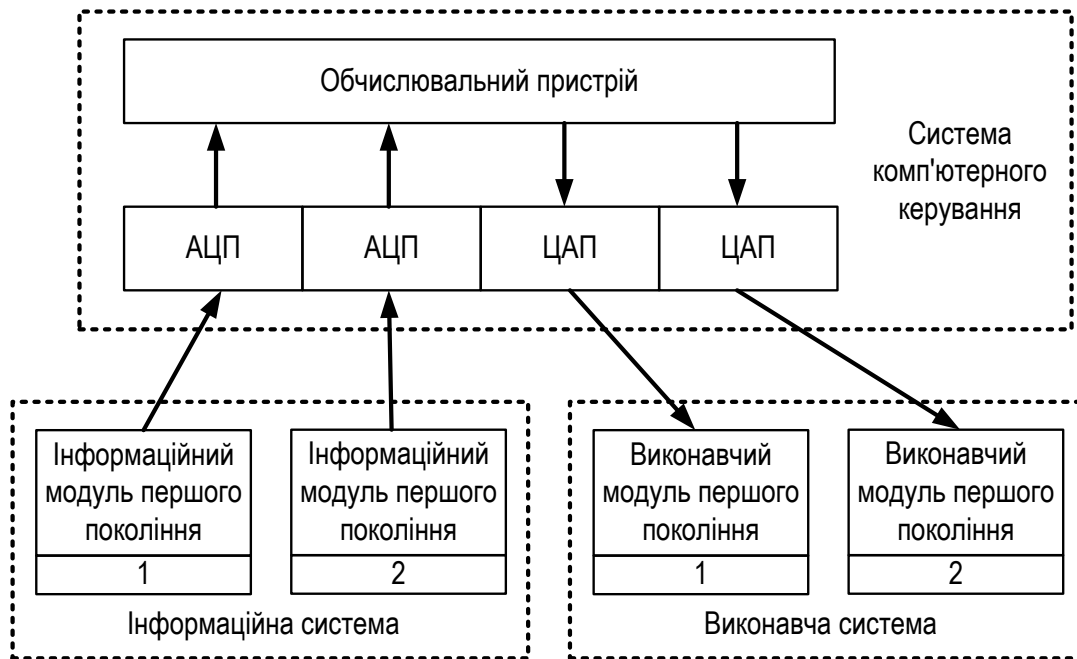


Рис. 3.10. Узагальнена структура мехатронного об'єкта з аналоговими інтерфейсами

Наступний клас інтерфейсів становлять цифрові дуплексні інтерфейси, прикладом яких, зокрема, є інтерфейси RS-232, RS-422, ARINC429. Використання таких інтерфейсів у системі комп'ютерного керування можливо у випадку застосування в складі мехатронного об'єкта інформаційних і виконавчих модулів другого покоління, що забезпечують обмін даними зі системою комп'ютерного керування в цифровому виді (рис. 3.11).

При цьому кожний інформаційний і виконавчий мехатронний модуль, що входить до складу мехатронного об'єкта, повинен підключатися до системи комп'ютерного керування за допомогою окремого контролера. На відміну від аналогового інтерфейсу цифровий дуплексний інтерфейс має велику перешкодозахищеність із меншими обмеженнями на дальність передачі даних, що забезпечує більш високу надійність. Однак, кабельна мережа при цьому не зменшується, оскільки знов-таки кожний мехатронний

модуль повинен бути пов'язаний із системою комп'ютерного керування окремою лінією зв'язку, тобто вартість буде досить високою. Тому цифрові дуплексні інтерфейси донедавна використовувалися при створенні системи комп'ютерного керування мехатронних об'єктів, до яких пред'являлися досить високі вимоги по надійності, але масогабаритні й вартісні характеристики були не настільки великі.

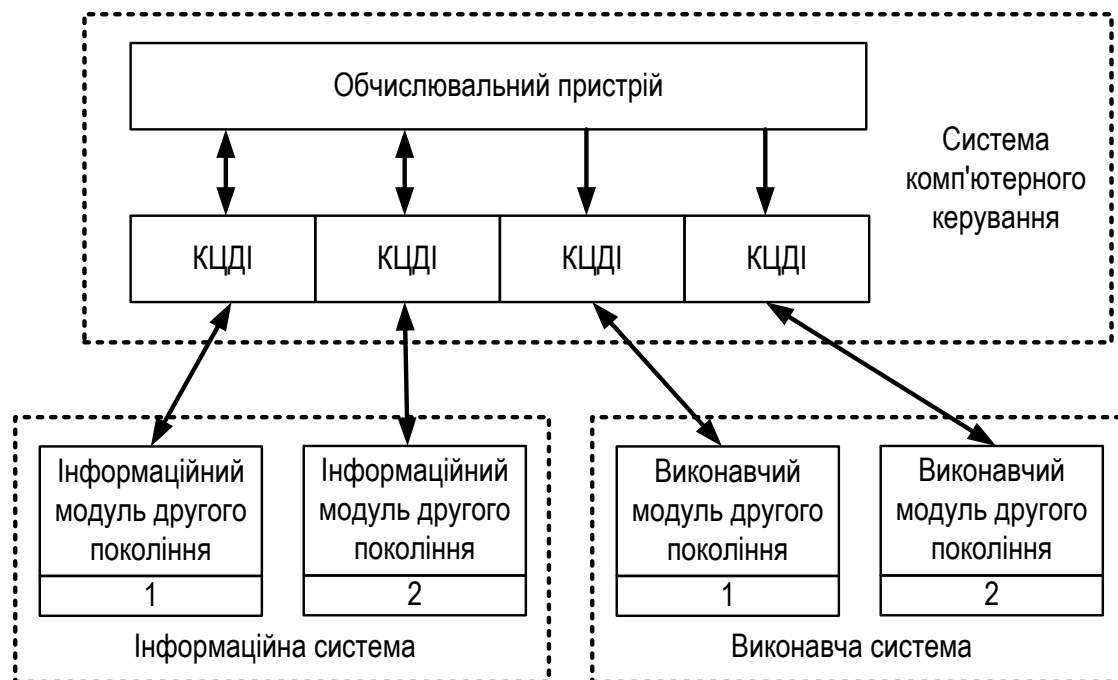


Рис. 3.11. Узагальнена структура мехатронного об'єкта із цифровим дуплексним інтерфейсом (КЦДІ – контролер цифрового дуплексного інтерфейсу)

Як приклад мехатронного об'єкта, який використовує цифровий дуплексний інтерфейс для зв'язку системи комп'ютерного керування з виконавчою й інформаційною системами, можна привести прискорювач часток ГЕЗА (рис. 3.6). Тут зв'язок між інформаційними й виконавчими мехатронними модулями з контролерами, які входять до складу системи комп'ютерного керування, здійснюється за допомогою цифрового дуплексного інтерфейсу RS-232C.

Нарешті, третій клас інтерфейсів становлять цифрові мережеві інтерфейси, які в цей час знаходять усе більш широке застосування при створенні сучасних складних мехатронних об'єктів. Узагальнена структура мехатронного об'єкта, що використовує інтерфейси такого типу, наведена на рис. 3.12.

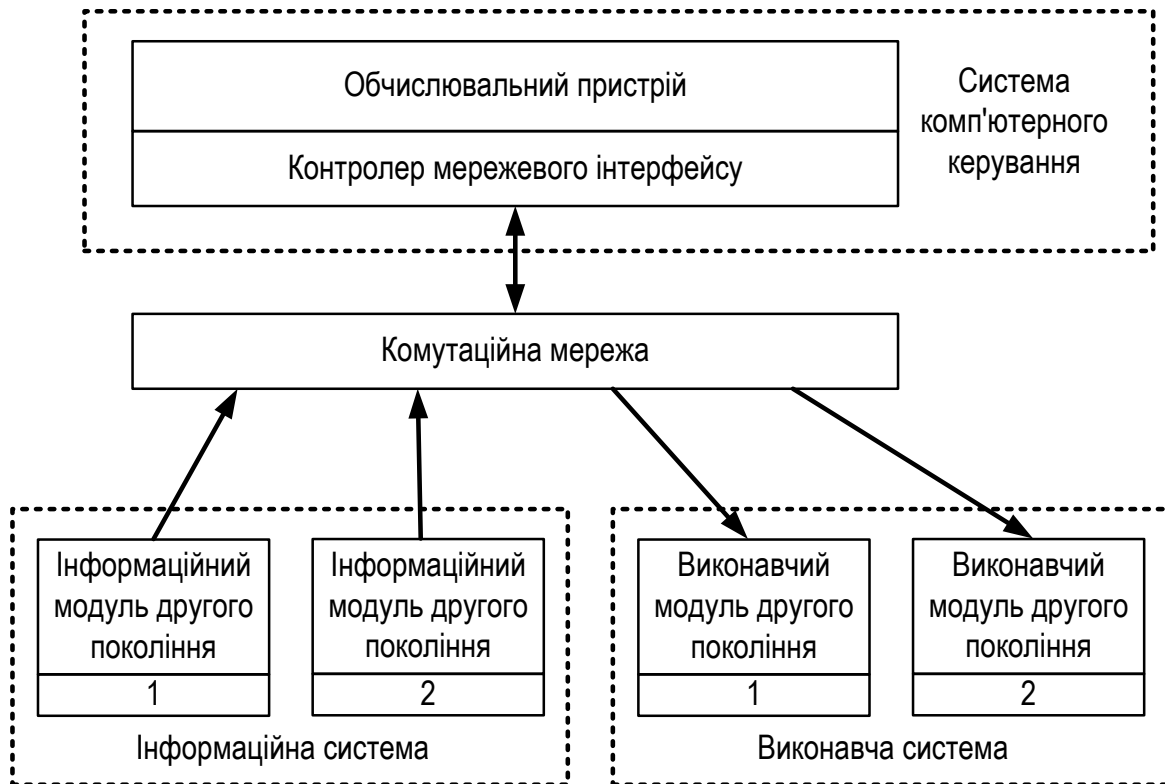


Рис. 3.12. Узагальнена структура мехатронного об'єкта з мережевим інтерфейсом

У цьому випадку обмін даними між системами комп'ютерного керування, виконавчою та інформаційною системами мехатронного об'єкта здійснюється за допомогою деякої комутаційної мережі (шини), до якої за допомогою контролера мережевого інтерфейсу підключається обчислювальний пристрій системи комп'ютерного керування. У якості прикладів подібної комутаційної мережі (шини) можна привести мережі типу MIL, STD, 1553B, CAN, Ethernet.

До переваг мережевої організації інтерфейсів мехатронного об'єкта слід віднести:

- високу перешкодозахищеність внаслідок використання цифрових даних;
- практична відсутність обмежень на дальність передачі даних;
- зменшення довжини кабельної мережі у порівнянні з аналоговим і цифровим дуплексним інтерфейсами, оскільки тут не треба прокладати окрему лінію зв'язку між кожним мехатронним модулем і системою комп'ютерного керування, а обмін даними здійснюється за допомогою загальної мережної шини.

Як приклад мехатронних об'єктів, що використовують мережний інтерфейс, можна привести верстат зі ЧПК та космічну станцію.

У таблиці 3.2 наведені якісні оцінки різних типів інтерфейсів системи комп'ютерного керування мехатронних об'єктів з погляду прийнятих вище критеріїв.

Таблиця 3.2. Типи інтерфейсів та їх характеристики

Тип інтерфейсу	Надійність	Вартість	Швидкодія
Аналоговий	Низька	Середня	Висока
Цифровий дуплексний	Висока	Висока	Висока
Цифровий мережевий	Висока	Середня	Середня

Накладення якісних оцінок різних способів організації обчислювальної та інтерфейсної частин системи комп'ютерного керування (таблиці 3.1 і 3.2) дозволяє здійснити якісну оцінку різних способів організації системи комп'ютерного керування в цілому, з якої випливає, що найбільш ефективним способом організації системи комп'ютерного керування мехатронного об'єкта з погляду критеріїв швидкодії, надійності й вартості є розподілена децентралізована організація обчислювального пристрою системи комп'ютерного керування з мережевим інтерфейсом. Надалі для стислості системи комп'ютерного керування з такою організацією будемо називати децентралізованою системою комп'ютерного керування.

Однак, незважаючи на очевидні переваги, практично відсутні приклади реального практичного застосування децентралізованої системи комп'ютерного керування в реальних мехатронних об'єктів. Це викликано, очевидно, двома обставинами:

- по-перше, відсутністю теоретично обґрунтованих методик їх проектування;
- по-друге, відсутністю алгоритмів децентралізованої організації обчислювального процесу при розв'язуванні завдань керування в подібних системи комп'ютерного керування [52].

3.4. Передача даних між мехатронними пристроями

Обмін інформацією між пристроями, що входять до складу мехатронної системи (комп'ютерами, контролерами, датчиками, виконавчими пристроями), відбувається в загальному випадку через промислову мережу (Fieldbus, „польову шину”) [53].

Промислові мережі відрізняються від офісних наступними властивостями:

- 1) спеціальними конструктивними виконаннями, що забезпечують захист від пилу, вологи, вібрації, ударів;
- 2) широким температурним діапазоном (звичайно від -40 до +70 °С);
- 3) підвищеною міцністю кабелю, ізоляції, роз'ємних з'єднань, елементів кріплення;
- 4) підвищеною стійкістю до впливу електромагнітних перешкод;
- 5) можливістю резервування для підвищення надійності;
- 6) підвищеною надійністю передачі даних;
- 7) можливістю самовідновлення після збою;
- 8) детермінованістю (визначеністю) часу доставки повідомлень;
- 9) можливістю роботи в реальному часі (з малою, постійною й відомою величиною затримки);

10) роботою з довгими лініями зв'язку (від сотень метрів до декількох кілометрів).

Промислові мережі звичайно не виходять за межі одного підприємства. Однак з появою Ethernet та Internet, для промислових мереж стали застосовувати ту ж класифікацію, що й для офісних:

- ▶ LAN (Local Area Network) - мережі, розташовані на обмеженій території (у цеху, офісі, у межах заводу);
- ▶ MAN (Metropolitan Area Network) - мережі міст;
- ▶ WAN (Wed Area Network) - глобальна мережа, що охоплює кілька міст або континентів. Звичайно для цього використовують Інтернет-технологію.

У цей час налічується більш 50 типів промислових мереж (Modbus, Profibus, DeviceNet, CANopen, LonWorks, ControlNet, SDS, Seriplex, ArcNet, BACnet, FDDI, FIP, FF, ASI, Ethernet, WorldFIP, Foundation Fieldbus, Interbus, BitBus). Однак широко розповсюдженими є тільки частина з них. В останні роки зріс інтерес до мереж на основі DeviceNet та CANopen [31].

3.4.1. Загальні відомості про промислові мережі

Промисловою мережею називають комплекс обладнання та програмного забезпечення, які забезпечують обмін інформацією (комунікацію) між декількома пристроями. Промислова мережа є основою для побудови розподілених систем збору даних і керування.

Оскільки мережні інтерфейси можуть бути невід'ємною частиною пристроїв, що з'єднуються, а мережне програмне забезпечення виконується на основному процесорі промислового контролера, то відокремити мережну частину від пристроїв, поєднаних у мережу, іноді фізично неможливо. З іншого боку, зміну однієї мережі на іншу часто можна виконати за допомогою заміни мережного програмного забезпечення й мережного адаптера або введенням перетворювача інтерфейсу, тому часто той самий тип

програмних логічних контролерів може використовуватися в мережах різних типів.

З'єднання промислової мережі з її компонентами (пристроями, вузлами мережі) виконується за допомогою інтерфейсів. Мережним інтерфейсом називають логічну й (або) фізичну границю між пристроєм і середовищем передачі інформації. Звичайно цією границею є набір електронних компонентів і пов'язаного з ними програмного забезпечення. При істотних модифікаціях внутрішньої структури пристрою або програмного забезпечення інтерфейс залишається без змін.

Найбільш важливими параметрами інтерфейсу є пропускна здатність і максимальна довжина кабелю, що підключається. Промислові інтерфейси звичайно забезпечують гальванічну розв'язку між пристроями, що з'єднуються. Найпоширеніші в промисловій автоматизації послідовні інтерфейси RS-422, RS-485, RS-232, Ethernet, CAN, HART та інші.

Для обміну інформацією взаємодіючі пристрої повинні мати однаковий протокол обміну. У найпростішій формі протокол - це набір правил, які керують обміном інформацією. Він визначає синтаксис і семантику повідомлень, операції керування, синхронізацію стану при комунікації. Протокол може бути реалізований апаратно, програмно або програмно-апаратно. Назва мережі звичайно збігається з назвою протоколу, що пояснюється його визначальною роллю при створенні мережі.

Звичайно мережа використовує кілька протоколів, що утворюють стек протоколів - набір зв'язаних комунікаційних протоколів. Для більшості мереж стек протоколів реалізований за допомогою спеціалізованих мережних мікросхем або вбудований в універсальний мікропроцесор.

Взаємодія пристроїв у промислових мережах виконується відповідно до моделей „клієнт-сервер” або „видавець-передплатник” (виробник - споживач) [54]. У моделі „клієнт-сервер” взаємодіють два об'єкти. Сервером є об'єкт, який надає сервіс, тобто виконує деякі дії по запиті клієнта. Мережа може містити кілька серверів і кілька клієнтів. Кожний клієнт може посилати

запити декільком серверам, а кожний сервер може відповідати на запити декількох клієнтів. Ця модель зручна для передачі даних, які з'являються періодично або в заздалегідь відомий час, як, наприклад, значення температури в періодичному технологічному процесі. Однак ця модель незручна для передачі випадково виникаючих подій, наприклад події, що полягає у випадковому спрацьовуванні датчика рівня, оскільки для одержання інформації про цю подію клієнт повинен періодично, з високою частотою, опитувати стан датчика й аналізувати його, перевантажуючи мережу трафіком, який не приносить користі.

У моделі взаємодії „видавець-передплатник” є один видавець і безліч передплатників. Передплатники повідомляють видавцю список тегів, значення яких вони прагнуть одержувати за певним розкладом або в міру появи нових даних. Кожний клієнт може підписатися на свій набір тегів. Відповідно до встановленого розкладу видавець розсилає передплатникам потрібну інформацію.

У будь-якій моделі взаємодії можна виділити пристрій, який керує іншим (підлеглим) пристроєм. Пристрій, що виявив ініціативу в обміні, називають ведучим, головним або майстром (Master). Пристрій, який відповідає на запити майстра, називають веденим, підлеглим або слейвом (Slave). Ведений пристрій ніколи не починає комунікацію першим. Він чекає запиту від ведучого й тільки відповідає на запити. Наприклад, у моделі клієнт-сервер клієнт є майстром, сервер - підлеглим. У моделі видавець-передплатник на етапі підписки майстром є клієнт, а на етапі розсилання публікацій - сервер.

У мережі може бути один або декілька ведучих пристроїв. Такі мережі називаються одномастерними або багатомастерними відповідно. У багатомастерній мережі виникає проблема розв'язання конфліктів між пристроями, що намагаються одночасно одержати доступ до середовища передачі інформації. Конфлікти можуть бути розв'язані методом передачі маркера, як, наприклад, у мережі Profibus, методом побітного порівняння

ідентифікатора (використовується в CAN), методом прослуховування мережі (використовується в Ethernet) і методом запобігання колізій (використовується в бездротових мережах).

У всіх мережах застосовується широкомовне розсилання без певної адреси, тобто всім учасникам мережі. Такий режим використовується звичайно для синхронізації процесів у мережі, наприклад для одночасного запуску процесу введення даних усіма пристроями або для синхронізації часу окремих пристроїв.

Деякі мережі використовують багатоабонентський режим, коли одне і те саме повідомлення надсилається декільком пристроям одночасно.

Передача інформації в мережі виконується через канал між передавальним і прийомним пристроєм. Канал є поняттям теорії інформації й містить у собі лінію зв'язку й пристрої для прийому і передачі. У загальному випадку замість терміну „лінія зв'язку” використовують термін „середовище передачі”, у якості якої може виступати, наприклад, оптоволокно, ефір або вита пара проводів.

У розподілених системах на основі промислових мереж може бути п'ять типів даних: сигнали, команди, стани, події, запити [55].

СИГНАЛИ - це результати вимірів, одержувані від датчиків і вимірювальних перетворювачів. Їх „час життя” дуже малий, тому часто потрібно одержати тільки останні дані й у максимально короткий строк.

КОМАНДИ - це повідомлення, які викликають деякі дії, наприклад, закриття клапана або включення ПІД-регулятора. Більшість систем повинні обробляти потоки КОМАНД, які передаються адресатові з високою надійністю і їх не можна передати повторно.

СТАН показує поточний або майбутній стан системи, у який вона повинна перейти. Вимога до часу доставки цих даних може бути не такою жорсткою, як для команд; неприйнятий СТАН може бути посланий повторно.

ПОДІЯ настає звичайно при досягненні поточним параметром граничного значення. Наприклад, ПОДІЄЮ може бути вихід температури за технологічно припустиму границю. За появою ПОДІЇ повинні йти відповідні дії, тому для ПОДІЙ особливо важлива вимога гарантованого часу доставки.

ЗАПИТ - це команда, яка посилається для того, щоб одержати відповідь. Прикладом може бути ЗАПИТ серверу, який видає на нього відповідь.

Мережі можуть мати топологію зірки, кільця, шини або змішану. „Зірка” у промисловій автоматизації використовується рідко. Кільце використовується в основному для передачі маркера в багатомасерних мережах. Шинна топологія є загальноприйнятою, що є однією із причин застосування терміна „промислова шина” замість „промислова мережа”. До загальної шини в різних місцях може бути підключена довільна кількість пристроїв.

Основними параметрами промислових мереж є продуктивність і надійність [56]. Продуктивність мережі характеризується часом реакції й пропускною здатністю.

Час реакції мережі визначається як інтервал часу між запитом ведучого пристрою й відповіддю веденого за умови, що ведений пристрій має таку малу затримку „вироблення” відповіді на запит, що нею можна знехтувати.

Пропускна здатність мережі визначає кількість інформації, яка переноситься мережею в одиницю часу. Вона вимірюється в бітах за секунду й залежить від швидкодії мережних приймачів, передавачів і середовища передачі.

Важливою характеристикою промислових мереж є надійність доставки даних. Надійність характеризується коефіцієнтом готовності, ймовірністю доставки даних, передбачуваністю часу доставки, безпекою, стійкістю до відмов. Коефіцієнт готовності дорівнює відношенню часу наробітку до настання відмови до суми часу наробітку до відмови й часу відновлення після відмови.

Ймовірність доставки даних визначається завадостійкістю каналу передачі й детермінованістю доступу до каналу. У безпроводних мережах ймовірність втрати пакетів при передачі набагато вища, ніж у тих, які використовують провідники. В мережах з випадковим методом доступу до каналу існує ймовірність того, що дані ніколи не будуть доставлені абонентові.

Час доставки даних в офісних мережах Ethernet є випадковою величиною, однак у промисловому Ethernet ця проблема вирішена застосуванням комутаторів.

Безпека - це здатність мережі захистити передані дані від несанкціонованого доступу.

Стійкість до відмов - це здатність мережі продовжувати функціонування при відмові деяких елементів. При цьому характеристики системи можуть погіршитися, але вона не втрачає працездатності. [31]

3.4.2. Провідникові мережі

3.4.2.1. Інтерфейси RS-485, RS -422 і RS -232

Інтерфейси RS -485 і RS -422 описані в стандартах ANSI EIA (асоціація електронної промисловості) / TIA (асоціація телекомунікаційної промисловості) - 485-A та EIA/TIA - 422. Інтерфейс RS-485 є одним із найпоширеніших у мехатронних системах. Його використовують промислові мережі Modbus, Profibus, DP, Interbus, LON і безліч нестандартних мереж. Зв'язане це з тим, що за всіма основними показниками даний інтерфейс є найкращим із усіх можливих при сучасному рівні розвитку технології. Основними його перевагами є:

- 1) двосторонній обмін даними всього по одній витій парі проводів;
- 2) робота з декількома трансиверами, підключеними до одної і тієї ж лінії, тобто можливість організації мережі;
- 3) велика довжина лінії зв'язку;

4) досить висока швидкість передачі даних.

Принципи побудови.

В основі побудови інтерфейсу RS-485 лежить диференціальний спосіб передачі сигналу, коли напруга, яка відповідає рівню логічної одиниці або нуля, відраховується не від „землі”, а вимірюється як різниця потенціалів між двома передавальними лініями: Data+ і Data- (рис. 3.13). При цьому напруга кожної лінії відносно „землі” може бути довільною, але вона не повинна виходити за діапазон -7...+12 В.

Приймачі сигналу є диференціальними, тобто сприймають тільки різницю між напругами на лінії Data+ і Data-. При різниці напруг не менше 200 мВ, до +12 В вважається, що на лінії встановлене значення логічної одиниці, при напрузі менше -200 мВ, до -7 В - логічного нуля. Диференціальна напруга на виході передавача у відповідності зі стандартом повинна бути не меншою 1,5 В, тому при порозі спрацьовування приймача 200 мВ перешкода (у тому числі спадання напруги на омичному опорі лінії) може мати розмах 1,3 В над рівнем 200 мВ. Такий великий запас необхідний для роботи на довгих лініях з великим омичним опором. Фактично саме цей запас по напрузі й визначає максимальну довжину лінії зв'язку (1200 м) при низьких швидкостях передачі (менше 100 кбіт/с).

Завдяки симетрії ліній відносно „землі” у них наводяться перешкоди, близькі за формою й величиною. У приймачі з диференціальним входом сигнал виділяється шляхом віднімання напруг на лініях, тому після віднімання напруга перешкоди виявляється рівною нулю. У реальних умовах, коли існує невелика асиметрія ліній і навантажень, перешкода усувається не повністю, але послабляється суттєво.

Для мінімізації чутливості лінії передачі до електромагнітних перешкод використовується вита пара проводів. Струми, що наводяться в сусідніх витках, внаслідок явища електромагнітної індукції (правило гвинта) виявляються спрямованими назустріч один одному й взаємно

компенсуються. Ступінь компенсації визначається якістю виготовлення кабелю й кількістю витків на одиницю довжини.

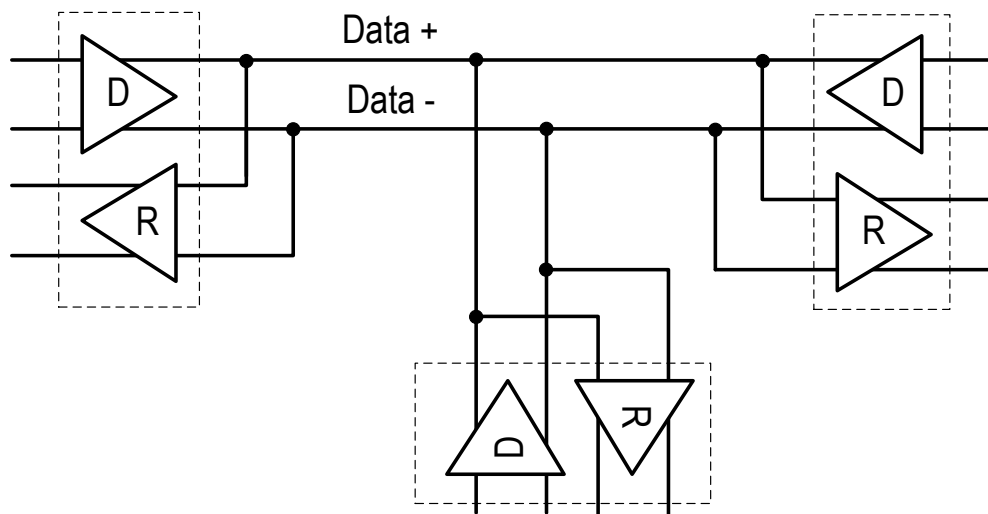


Рис. 3.13. З'єднання трьох пристроїв з інтерфейсом RS -485 по двопровідній схемі

Інтерфейс RS-485 має дві версії: двопровідну й чотирипровідну. Двопровідна використовується для напівдуплексної передачі (рис. 3.13), коли інформація може передаватися в обох напрямках, але в різний час. Для повнодуплексної (дуплексної) передачі використовують чотири лінії зв'язку: по двох інформація передається в одному напрямку, по двох інших - у зворотному (рис. 3.14).

Недоліком чотирипровідної (рис. 3.14) схеми є необхідність жорсткого визначення ведучого та веденого на стадії проектування системи, у той час як у двопровідній схемі будь-який пристрій може бути як у ролі ведучого, так і веденого. Перевагою чотирипровідної схеми є можливість одночасної передачі й приймання даних, що буває необхідно при реалізації деяких складних протоколів обміну.

Захист інтерфейсу від блискавки виконується за допомогою газорозрядних і напівпровідникових пристроїв захисту.

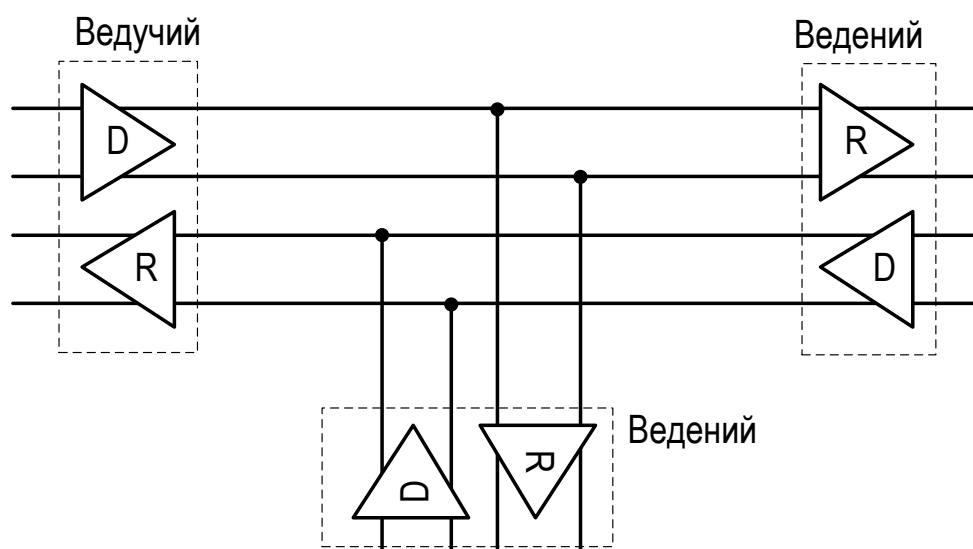


Рис. 3.14. Чотирипроводне з'єднання пристроїв з інтерфейсом RS-485

Топологія мережі на основі інтерфейсу RS-485

Топологія мереж на основі інтерфейсу RS-485 визначається необхідністю усунення явища відбиття у лінії передачі. Оскільки відбиття походять від будь-якої неоднорідності, у тому числі відгалужень від лінії, то єдино правильною топологією мережі буде така, яка виглядає як єдина лінія без відводів, до якої не більш ніж в 32 точках підключені пристрої з інтерфейсом RS-485 (рис. 3.15, а). Будь-які варіанти, у яких лінія має довгі відводи або з'єднання декількох кабелів в одній точці (рис. 3.15, б), приводять до відбиття і зниження якості передачі.

Однак сказане справедливо тільки для високих швидкостей передачі (більш 9600 біт/с), коли ефекти відбиття впливають на вірогідність передачі. Для низьких швидкостей довжина відводу може бути довільною.

Якщо існує необхідність розгалуження лінії, то це можна зробити за допомогою повторювачів інтерфейсу або концентратора (хаба). Повторювачі дозволяють розділити лінію на сегменти, у кожному з яких виконуються умови узгодження за допомогою двох термінальних резисторів і не виникають ефекти, пов'язані з відбиттями від кінців лінії, а довжина відводу від лінії до повторювача завжди може бути зроблена досить малою.

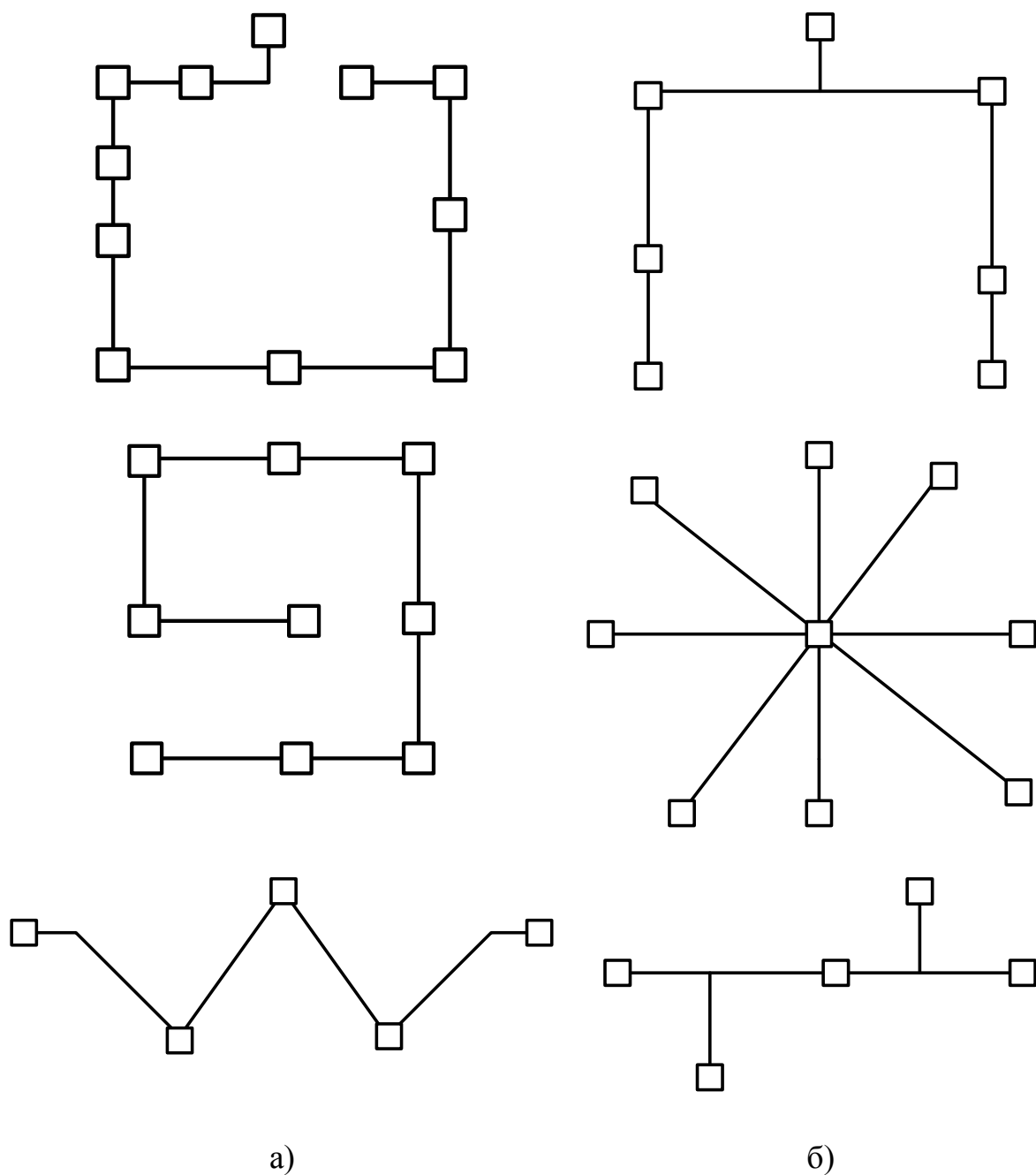


Рис. 3.15. Правильна (а) і неправильна (б) топології мережі на основі інтерфейса RS-485 (квадратиками позначені пристрої з інтерфейсом RS-485)

Розширення граничних можливостей

Стандарт RS-485 допускає підключення не більш 32 приймачів до одного передавача. Ця величина обмежується потужністю вихідного каскаду передавача при стандартному вхідному опорі приймача 12 кОм. Кількість навантажень (приймачів) може бути збільшена за допомогою потужніших передавачів, приймачів з більшим вхідним опором і проміжних

ретрансляторів сигналу (повторювачів інтерфейсу). Усі ці методи використовуються на практиці, коли це необхідно, хоча вони виходять за рамки вимог стандарту.

У деяких випадках потрібно з'єднати пристрої на відстані більш 1200 м або підключити до однієї мережі більше 32 пристроїв. Це можна зробити за допомогою повторювачів (ретрансляторів) інтерфейсу. Повторювач встановлюється між двома сегментами лінії передачі, приймає сигнал одного сегмента, відновлює фронти імпульсів і передає його за допомогою стандартного передавача в другий сегмент (рис. 3.16). Такі повторювачі звичайно є двонаправленими й мають гальванічну ізоляцію.

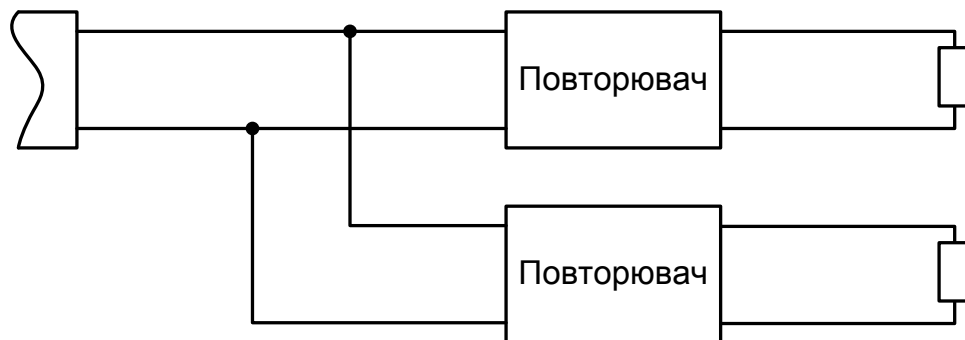


Рис. 3.16. Використання повторювачів інтерфейсу для розгалуження лінії передачі

Розповсюдженим методом збільшення числа навантажень лінії є використання приймачів з більш високоомним входом, аніж це передбачено стандартом EIA/TIA-485 (12 кОм). Наприклад, при входному опорі приймача 24 кОм до стандартного передавача можна підключити 64 приймача. Уже випускаються мікросхеми трансиверів для інтерфейсу RS-485 з можливістю підключення 64, 128 і 256 приймачів в одному сегменті мережі. Відзначимо, що збільшення кількості навантажень шляхом збільшення входного опору приймачів приводить до зменшення потужності переданого по лінії сигналу й, як наслідок, до зниження завадостійкості.

Інтерфейси RS -232 і RS -422

Інтерфейс RS-422 використовується набагато рідше, ніж RS-485 і, як правило, не для створення мережі, а для з'єднання двох пристроїв на великій відстані (до 1200 м), оскільки інтерфейс RS-232 працездатний тільки на відстані до 15 м. Кожний передавач RS-422 може бути навантажений на 10 приймачів. Інтерфейс працездатний при напрузі до ± 7 В. У табл. 3.3 наведено порівняння основних характеристик трьох найбільш популярних інтерфейсів, використовуваних у промисловій автоматизації.

Таблиця 3.3. Порівняння інтерфейсів RS-232, RS-422, RS-485

Параметр	RS-232	RS-422	RS-485
Максимальна кількість приймачів	1	10	32
Максимальна швидкість передачі, Мбіт/с	0,46	10	30*
Напруга в лінії під навантаженням, В	$\pm 5 \dots \pm 15$	± 2	$\pm 1,5$
Допустимий діапазон сигналів на вході приймача, В	± 15	± 10	$-7 \dots +12$

* Швидкість передачі даних 30 Мбіт/с забезпечується сучасною елементною базою, але вона є нестандартною [31].

3.4.2.2. CAN

CAN (Control Area Network) - послідовна магістраль, що забезпечує з'єднання в мережу „інтелектуальних” пристроїв введення/виводу, датчиків і виконавчих пристроїв деякого механізму або навіть підприємства. Характеризується протоколом, який забезпечує можливість знаходження на магістралі декількох ведучих пристроїв, що забезпечують передачу даних у реальному масштабі часу й корекцію помилок, високою завадостійкістю. Система CAN забезпечена великою кількістю мікросхем, що забезпечують роботу підключених до магістралі пристроїв, розробку яких починала фірма BOSCH для використання в автомобілях, і в цей час широко використовуваних в мехатронних системах, зокрема у автомобільній промисловості. Основні характеристики CAN-інтерфейсу представлені у табл. 3.4.

Таблиця 3.4. Основні параметри CAN-інтерфейсу

Параметр	Значення параметру
Швидкість передачі, Мбіт/с	1 (максимум)
Відстань передачі, м	1000 (максимум)
Характер сигналу	Диференціальна напруга
Лінія передачі	Вита пара
Кількість приймачів	64
Схема з'єднання	напівдуплекс, багатоточкова

CAN-інтерфейс призначений для організації високонадійних недорогих каналів зв'язку в розподілених системах керування. Інтерфейс широко застосовується в промисловості, енергетиці й на транспорті. Він дозволяє будувати дешеві канали зв'язку і високошвидкісні мережі.

Швидкість передачі задається програмно. Користувач вибирає швидкість, виходячи з відстаней, числа абонентів і ємності ліній передачі (табл. 35.)

Таблиця 3.5. Відповідність між відстанню та швидкістю передачі даних

Відстань, м	25	50	100	250	500	1000	2500	5000
Швидкість, Кбіт/с	1000	800	500	250	125	50	20	10

Максимальне число абонентів, підключених до даного інтерфейсу фактично визначається навантажувальною здатністю застосованих приймачів та передавачів. Наприклад, при використанні трансивера фірми PHILIPS PCA82C250 вона рівна 110.

Протокол CAN використовує оригінальну систему адресації повідомлень. Кожне повідомлення забезпечується ідентифікатором, який визначає призначення переданих даних, але не адресу приймача. Будь-який приймач може реагувати як на один ідентифікатор, так і на декілька. На один ідентифікатор можуть реагувати кілька приймачів.

Протокол CAN має розвинену систему виявлення й сигналізації помилок. Загальна ймовірність невиявленої помилки $4,7 \times 10^{-11}$. Система арбітражу протоколу CAN виключає втрату інформації й часу при „зіткненнях” на шині.

Інтерфейс із застосуванням протоколу CAN легко адаптується до фізичного середовища передачі інформації. Це може бути диференціальний сигнал, оптоволокно, просто відкритий колектор тощо. Нескладно виконується гальванічна розв’язка. Елементна база, що підтримує CAN, широко випускається в індустріальному виконанні [31].

3.4.3. Безпроводні мережі

3.4.3.1. Області використання та переваги безпроводних мереж

Існує багато мехатронних систем, де складно обійтися без безпроводних мереж або де їх застосування бажане:

- ▶ датчики й виконавчі пристрої на рухомих частинах конвеєрів, вітряних млинів, ліфтів, міксерів, візків для переміщення вантажів по цеху, на крилах літаків, на підшипниках двигунів, на роботах, у пересувних лабораторіях, датчики на тілі людей та тварин;
- ▶ об’єкти, у яких небажано свердлими стіни або псувати дизайн: офісні будинки, у яких установлюється пожежна й охоронна сигналізація, датчики для систем обігріву й кондиціонування повітря, для моніторингу механічних напруг у конструкціях будинків; у системах „розумного будинку” (керування освітленням приміщення, кондиціонуванням, обігрівом, охоронними датчиками, побутовими приладами тощо);
- ▶ епізодичне програмування й діагностика ПЛК, коли прокладати постійні кабелі не вигідно;
- ▶ дистанційне зчитування показань лічильників, самописців;

- ▶ об'єкти з агресивними середовищами, вібрацією; об'єкти, що перебувають під високою напругою або в місцях, не зручних для прокладки кабелю;
- ▶ відстеження траєкторії руху транспорту, моніторинг сільськогосподарських культур, моніторинг шкідливих викидів тощо;
- ▶ будь-які об'єкти, для яких відомо, що вартість кабелів, кабельних каналів, опор та інших пристроїв, а також робіт з монтажу й обслуговування суттєво перевищує вартість безпроводної системи, що їх заміняє, за умови відсутності жорстких вимог до надійності доставки повідомлень у реальному часі;
- ▶ об'єкти у вибухонебезпечних зонах.

У більшості застосувань безпроводні мережі дозволяють досягти наступних переваг у порівнянні із провідниковими мережами [57]:

- 1) суттєво знижується вартість установки датчиків;
- 2) виключається необхідність профілактичного обслуговування кабелів;
- 3) виключаються дорогі місця у розгалуженні кабелів;
- 4) зменшується кількість кабелів;
- 5) зменшуються витрати праці й часу на монтаж і обслуговування системи;
- 6) знижується вартість системи за рахунок виключення кабелів;
- 7) знижується вимоги до навчання персоналу монтажної організації;
- 8) прискорюється налагодження системи й пошук несправностей;
- 9) забезпечується зручна модернізація системи.

Оскільки реконфігурація системи і її монтаж стають набагато більш простими, безпроводні мережі можна використовувати й у традиційних областях застосування кабельних зв'язків, коли вартість кабелю й монтажу виявляється вищою, ніж установка безпроводної системи.

Найбільше поширення одержали три типи безпроводних мереж: Bluetooth [58], Wi-Fi [59] та ZigBee [60]. Протоколи цих мереж розроблені й підтримуються організаціями, які мають такі самі назви. Тому в назві мереж

звичайно вказують посилання на стандарт. Усі три мережі використовують частотний діапазон, який не має ліцензії 2,4 ГГц [31].

3.4.3.2. Проблеми безпроводних мереж і шляхи їх розв'язання

З погляду вимог до промислових мереж безпроводні мережі уступають провідниковим за наступними показниками:

- ▶ час доставки повідомлень: використовуваний механізм випадкового доступу до каналу не гарантує доставку пакета інформації в заздалегідь відомий час [61] і цю проблему не можна розв'язати за допомогою комутаторів, як у провідникових мережах;
- ▶ перешкодозахищеність: безпроводні мережі піддаються впливу електромагнітних перешкод значно сильніше, ніж провідникові;
- ▶ надійність зв'язку: зв'язок може зникнути при несвоєчасній заміні батарей живлення, заміні розташування вузлів мережі або появі об'єктів, що вносять загасання, відбиття, переломлення або розсіювання радіохвиль;
- ▶ обмежена дальність зв'язку без використання ретрансляторів (звичайно не більш 100 м усередині приміщень);
- ▶ різке падіння пропускної здатності мережі при збільшенні кількості одночасно працюючих станцій і коефіцієнту використання каналу;
- ▶ безпека: можливість витоку інформації, незахищеність від штучно створених перешкод, можливість непомітного керування технологічним процесом іншими особами.

Найбільшою перевагою безпроводних мереж є відсутність кабелів, що й визначає вибір областей їх застосування в системах мехатроніки.

Розглянемо фізичні причини виникнення деяких із перерахованих проблем і методи боротьби з ними. Основними причинами проблем без провідникових мереж є інтерференція, дифракція, переломлення, відбиття, розсіювання і зниження потужності випромінювання при збільшенні відстані

від джерела, а також неможливість локалізації радіохвиль в обмеженому просторі.

Залежність потужності сигналу від відстані. Відомо, що потужність радіохвилі зменшується по мірі віддалення від антени внаслідок розходження пучка, розсіювання й поглинання хвиль перешкодами на шляху їх поширення. Кожний учасник безпроводної мережі має обмежену зону гарантованого приймання, яка представляється приблизно у формі сфери. Це призводить до необхідності планування розташування безпроводних станцій таким чином, щоб зони гарантованого приймання станцій, що безпосередньо зв'язуються, перекривалися. Якщо станції розташовані на відстані негарантованого приймання, то невеликі зміни навколишнього оточення можуть призвести до втрати повідомлень або зниження швидкості передачі.

Обмеженість радіуса дії передавачів привела до виникнення коміркових мереж [62], у яких інформація передається не через загальний канал зв'язку, як у провідникових мережах, а від вузла до вузла, використовуючи проміжні вузли мережі як ретранслятори і маршрутизатори. При виході з ладу або видаленні з мережі деяких вузлів мережа автоматично знаходить новий маршрут, щоб доставити дані адресатові. Додавання до мережі нового пристрою також може відбуватися автоматично, тобто коміркові мережі мають властивість самоорганізації.

Джерела перешкод. Існує декілька причини спотворення переданого сигналу: паразитний взаємовплив сусідніх каналів, ефект Доплера, перешкоди від працюючих двигунів, розряди статичної електрики тощо. Це може привести до втрати пакета, повторній передачі та, як наслідок, непередбаченій затримці в каналі. Інтенсивність потоку помилок залежить від потужності джерел перешкод, типу модуляції й потужності передавача, його частотного діапазону та інших причин.

Наслідком перешкод у каналі може бути не тільки втрата даних або затримка передачі, але й „проблема просторової неузгодженості”. Вона полягає в наступному. Коли система використовує ширококомовний режим

передачі без повідомлення про одержання, передбачається, що всі приймачі повинні одержати ті самі дані одночасно. Однак внаслідок помилок у каналі деякі споживачі можуть одержати помилкові дані. Така помилка особливо небажана, якщо ширококомовний режим використовується для забезпечення синхронної роботи декількох контролерів у тому самому технологічному процесі, оскільки вона приведе до розсинхронізації процесу.

Особливістю розглянутого випадку є те, що ймовірність помилки в системі різко зростає в порівнянні з імовірністю помилки в одному каналі p . Оскільки ймовірність безпомилкової передачі в системі є добутком ймовірностей безпомилкової передачі в кожному з каналів, то при кількості однакових каналів k ймовірність відсутності збоїв у системі буде рівна $(1-p)^k$. Наприклад, у системі з 8 каналів при ймовірності помилки в каналі $p=0,1$ імовірність безпомилкової передачі складе всього 43 %.

Одним із прикладів, де описана ситуація може відігравати негативну роль, є режим одночасного введення декількома пристроями сигналів датчиків. У провідникових мережах для цього використовують ширококомовні команди, які доходять до всіх пристроїв одночасно. Якщо аналогічний режим використовувати в безпроводній мережі, то ймовірність того, що всі k датчиків стануть тимчасово недоступні одночасно буде також рівна $(1-p)^k$.

Інші особливості безпроводних каналів. Ряд особливостей безпроводної передачі даних не дозволяє використовувати багато методів, характерних для провідникових мереж.

Безпроводні трансивери не можуть передавати й приймати сигнал на тому самому каналі. Це пов'язано зі швидким зменшенням потужності випромінювання при збільшенні відстані. Сигнал власного передавача виявляється на порядки сильнішим прийнятого сигналу й заглушає його. На відміну від цього, у провідникових каналах обидва сигнали мають приблизно однакову потужність. Тому безпроводні трансивери в принципі не можуть прослуховувати лінію під час передачі, як це робиться, наприклад, у CAN або Ethernet.

Наступною проблемою є електроживлення безпроводних мереж, оскільки провідникові пристрої (в основному датчики) не повинні мати проводів (підключатися куди-небудь за допомогою проводів). Тому дуже актуальна проблема економії енергії батарей, пошук простих способів їх заміни, виключення відмов через розряд, пошук альтернативних джерел енергії. У літературі розглядаються варіанти передачі енергії живлення електромагнітними хвилями, трансформаторами з великим повітряним зазором (на відстань до декількох метрів), добування енергії згоряння палива, застосування сонячних батарей.

Методи зменшення кількості помилок у каналі. Отже, помилки в радіоканалі з'являються частіше, ніж в екранованому кабелі, і звичайно, носять характер збоїв, у той час як у провідникових системах, навпаки, помилки частіше бувають катастрофічними, пов'язаними з виходом з ладу (відмовою) каналного обладнання. Проте, після прийняття всіх заходів потік помилок у радіоканалі може бути знижений до необхідного рівня. Наприклад, в авіації ймовірність помилок у безпроводних каналах становить менше 10^{-19} [63]. Однак такі значення ймовірності досягаються дуже великими зусиллями. Тому найбільш перспективною областю застосування безпроводних мереж є системи, у яких допускається деякий відсоток помилок. Ймовірність помилки може бути використана як компонент цільової функції при проектуванні безпроводної системи.

У промислових мережах часто використовується режим ширококомовної передачі, коли повідомлення одночасно повинні прийняти всі учасники мережі. Його особливістю є відсутність підтвердження про одержання повідомлення. У силу низької ймовірності безпомилкової передачі по радіоканалу для реалізації ширококомовної передачі необхідно вжити заходів для збільшення ймовірності доставки повідомлень у безпроводному каналі. Одним з можливих методів є кодування ширококомовного повідомлення з великою надмірністю, при якій приймач може відновити загублені під час

передачі біти. Незважаючи на зниження пропускну здатності каналу, такий метод може бути дуже ефективним.

Для збільшення вірогідності передачі використовують метод ARQ (Automatic Repeat Request - автоматичний повтор у відповідь на запит). Метод ARQ може використовувати, наприклад, такі принципи:

- ▶ передача додатково до повідомлення коригувального коду з великою надмірністю;
- ▶ відправлення одночасно декількох однакових пакетів;
- ▶ використання декількох антен для повторної передачі повідомлень.

Один з варіантів збільшення вірогідності передачі даних полягає в застосуванні декількох антен для одного приймача. Аналогічний варіант із декількома антенами може бути використаний для передавача. У методі передачі зі зворотним зв'язком передавальні антени перебираються по черзі, поки від приймача не прийде сигнал про те, що повідомлення прийняте. Якщо застосування декількох антен неможливо, замість додаткових антен можна використовувати інші станції як ретранслятори.

Питання безпеки. Проблеми безпеки розділяються на завдання аутентифікації (встановлення особистості), яка виконується звичайно за допомогою ідентифікації імені користувача й пароля, завдання розмежування прав доступу до системи й завдання захисту інформації за допомогою методів шифрування. Механізми шифрування [64] засновані на алгоритмах, які перетворюють сигнали, що несуть інформацію, у шумоподібні (псевдовипадкові) сигнали [31].

3.4.3.3. Bluetooth

У цей час існує три широко розповсюджені стандарти на безпроводникові мережі: Bluetooth, Zig-Bee і Wi-Fi. Пристрої для цих мереж не вимагають одержання ліцензії (що в багатьох випадках важливо), хоча й вимагають реєстрації.

Bluetooth був спроектований на базі стандарту IEEE 802.15.1 спеціально для заміни кабеля при з'єднанні різних пристроїв офісної й побутової техніки з використанням частотного діапазону 2,4 ГГц [58]. Специфікація Bluetooth підтримується організацією SIG (Bluetooth Special Interest Group), яка утворена в 1998 р., що включає 1900 членів. У системах автоматизації Bluetooth зручний для запису програм у ПЛК, дистанційного зчитування показників з накопичувачів інформації. Він організований у вигляді „пикомереж” (piconet), у яких один ведучий пристрій здійснює взаємодію не більш ніж із сьома веденими. Ведені пристрої можуть взаємодіяти один з одним тільки через ведучий. Кожний пристрій може бути членом чотирьох пикомереж одночасно, але головним може бути тільки в одній з них. Такий пристрій виконує роль моста між пикомережами. Декілька взаємодіючих пикомереж утворюють так звану scatternet („розкидану мережу”).

Трафік у мережі організований з тимчасовим поділом каналів і дуплексною передачею. Тимчасовий поділ здійснюється інтервалами (тимчасовими слотами) довжиною в 625 мкс. Ведучі пристрої можуть починати передачу тільки протягом інтервалів з непарними номерами, ведені відповідати протягом парних інтервалів. Протягом кожного інтервалу можна передати 366 біт.

В Bluetooth використовується широкосмугова модуляція типу FHSS. Перехід з однієї частоти на іншу виконується за випадковим законом, який встановлюється для кожного з'єднання індивідуально. Це підвищує ступінь захисту інформації. Несуча частота змінюється 1600 раз у секунду. Швидкість передачі рівна 433,9 кбіт/с.

Якщо пикомережі розташовані близько одна від одної, то вони можуть впливати одна на одну, оскільки між ними немає ніякої синхронізації. Щоб зменшити ймовірність взаємовпливу, використовується адаптивний метод стрибкоподібної зміни частоти AFH.

Кожний Bluetooth-пристрій має 48-бітову адресу. Більшість Bluetooth-пристроїв мають потужність передавача 1 мВт, однак дозволений наступний ряд потужностей, що ділить усі пристрої на три класи:

- ▶ клас 1 - до 100 мВт (максимальна дальність на відкритому просторі до 100 м);
- ▶ клас 2 - до 2,5 мВт (максимальна дальність на відкритому просторі до 15 м);
- ▶ клас 3 - до 1 мВт (максимальна дальність на відкритому просторі до 5 м).

Можна назвати наступні переваги технології Bluetooth: малі розміри обладнання, простота використання, безпека передачі інформації (завдяки аутентифікації та кодуванню), гарна підтримка стандартів. До недоліком можна віднести відносно велике споживання енергії й неможливість побудови мереж складної конфігурації. Ці особливості пов'язані з тим, що Bluetooth вирішує проблему заміни кабелів для пристроїв, що підключаються до комп'ютера, а не проблему створення безпроводної мережі [31].

3.4.3.4. Wi-Fi

Основне призначення технології Wi-Fi [59] (Wireless Fidelity – „бездротова точність”) – безпроводникове розширення мереж Ethernet. Вона використовується також там, де небажано або неможливо використовувати провідникові мережі, наприклад, для передачі інформації від рухомих частин механізмів; там, де не можна свердлити стіни; на великому складі, де комп'ютер потрібно носити із собою.

Wi-Fi розроблений консорціумом Wi-Fi на базі серії стандартів IEEE802.11 (1997 р.) і забезпечує швидкість передачі від 1...2 до 54 Мбіт/с. Wi-Fi-консорціум розробляє прикладні специфікації для втілення стандарту Wi-Fi у життя, виконує тестування й сертифікацію продукції інших фірм на відповідність стандарту, організує виставки, забезпечує необхідною інформацією розроблювачів Wi-Fi-обладнання.

Незважаючи на те що стандарт IEEE 802.11 був ратифікований ще в 1997 р., мережі Wi-Fi одержали широке поширення тільки в останні роки, коли суттєво знизилися ціни на серійне мережне обладнання. У промисловій автоматизації з безлічі стандартів серії 802.11 використовуються тільки два: 802.11b зі швидкістю передачі до 11 Мбіт/с і 802.11g (до 54 Мбіт/с).

Wi-Fi використовує метод доступу до мережі, у якому для зниження ймовірності виникнення конфліктів, помилок та інших неузгодженостей (так званих колізій) використані наступні принципи:

- ▶ перш ніж станція почне передачу, вона повідомляє, як довго вона буде займати канал зв'язку;
- ▶ станція не може почати передачу, поки не мине зарезервованій раніше час;
- ▶ учасники мережі не знають, чи прийнятий їх сигнал, поки не одержать підтвердження про це;
- ▶ якщо дві станції почали працювати одночасно, вони зможуть довідатися про це тільки по тому факту, що не одержать підтвердження про приймання;
- ▶ якщо підтвердження не отримане, учасники мережі вичікують випадковий проміжок часу, щоб почати повторну передачу.

Дальність зв'язку засобами Wi-Fi сильно залежить від умов поширення електромагнітних хвиль, типу антени й потужності передавача. Типові значення, що вказуються виготовлювачами обладнання, становлять 100-200 м у приміщенні й до декількох кілометрів на відкритій місцевості із застосуванням зовнішньої антени й при потужності передавача 50-100 мВт. Разом з тим, за повідомленням німецького щотижневика Computerwoche під час змагань по дальності зв'язку був зафіксований зв'язок на відстані 89 км із застосуванням стандартного обладнання стандарту IEEE 802.11b (2,4 ГГц) і супутникових антен („тарілок”). У книзі рекордів Гіннеса зафіксований також Wi-Fi-зв'язок на відстані 310 км із застосуванням антен, піднятих на велику висоту за допомогою повітряних куль.

Архітектура мережі Wi-Fi Стандарт IEEE 802.11 встановлює три варіанти топології мереж: незалежні базові зони обслуговування (Independent Basic Service Sets - IBSS); базові зони обслуговування (Basic Service Sets - BSS); розширені зони обслуговування (Extended Basic Service Sets - EBSS).

Під зоною обслуговування тут розуміється набір логічно згрупованих пристроїв. Кожна зона обслуговування має свій ідентифікатор (Service Set Identifier - SSID). Станція-приймач використовує SSID для визначення того, з якої зони обслуговування приходить сигнал.

В архітектурі IBSS станції зв'язуються безпосередньо одна з одною без використання точки доступу й без можливості приєднання до провідникової локальної мережі. Зона обслуговування IBSS використовується звичайно для об'єднання в мережу малої кількості станцій, оскільки в ній не передбачена можливість ретрансляції сигналу для збільшення дальності зв'язку.

При використанні BSS станції спілкуються одна із одною через загальний центральний вузол зв'язку, що називається точкою доступу. Точка доступу звичайно підключається до провідникової локальної мережі Ethernet.

Розширена зона обслуговування виходить при об'єднанні декількох BSS у єдину систему за допомогою розподільної системи, у якості якої може виступати провідна мережа Ethernet [31].

3.4.4. Методи перевірки помилок у отриманих даних

Контроль парності. Можливо, найпростіша форма виявлення помилок полягає у використанні одного біта парності. Припустимо, що на рис. 3.17 передані дані мають довжину d розрядів. При перевірці на парність відправник просто додає до даних один біт, значення якого обчислюється як сума всіх d розрядів даних по модулю 2. У цьому випадку кількість одиниць у числі, що виходить у результаті, завжди буде парною. Застосовуються також схеми, у яких контрольний біт інвертується, у результаті чого кількість одиниць у числі, що виходить у результаті, завжди буде непарною. На рис.

3.17 зображена схема перевірки на парність, а єдиний біт парності зберігається в окремому полі.



Рис. 3.17. Пакет даних до яких додається біт парності

Дії, що виконуються одержувачем при використанні такої схеми, також дуже прості. Одержувач повинен усього лише порахувати кількість одиниць в отриманих $d+1$ розрядах. Якщо при перевірці на парність одержувач виявляє, що в прийнятих ним даних непарна кількість одиничних розрядів, він розуміє, що відбулася помилка, щонайменше, в одному розряді. У загальному випадку це означає, що в отриманих даних інвертована непарна кількість розрядів (відбулася помилка непарної кратності).

Що відбудеться, якщо в отриманому пакеті даних відбудеться парна кількість однобітових помилок? У цьому випадку одержувач не зможе виявити помилку. Якщо ймовірність помилки в одному розряді мала й можна припустити, що помилки в окремих розрядах виникають незалежно одна від одної, тоді ймовірність декількох помилок в одному пакеті вкрай мала. У такому випадку єдиного біта парності може бути досить. Однак практичні спостереження показали, що в дійсності помилки не є незалежними, а часто групуються в пакети помилок. У випадку пакетних помилок імовірність того, що одержувач не виявить помилку в пакеті, може наблизитися до величини 50 %. Очевидно, у такій ситуації потрібно більш надійна схема виявлення помилок.

Контрольна сума це деяке значення, розраховане з набору даних шляхом застосування певного алгоритму, який використовується для перевірки цілісності даних при їхній передачі або зберіганні. Також контрольні суми можуть використовуватися для швидкого порівняння двох

наборів даних на нееквівалентність: з великою ймовірністю різні набори даних будуть мати різні контрольні суми. Це може бути використано, наприклад, для детектування комп'ютерних вірусів. Незважаючи на свою назву, контрольна сума не обов'язково обчислюється шляхом підсумовування.

З погляду математики контрольна сума є хеш-функцією, що використовується для обчислення контрольного коду - невеликої кількості біт (або байт) усередині великого блоку даних, наприклад, мережного пакета або блоку комп'ютерного файлу, який застосовується для виявлення помилок при передачі або зберіганні інформації. Значення контрольної суми додається в кінець блоку даних безпосередньо перед початком передачі або запису даних на який-небудь носій інформації. Згодом воно перевіряється для підтвердження цілісності даних.

Популярність використання контрольних сум для перевірки цілісності даних обумовлена тим, що подібна перевірка просто реалізується у двійковому цифровому обладнанні, легко аналізується й добре підходить для виявлення загальних помилок, викликаних наявністю шуму в каналах передачі даних.

Циклічний надлишковий код (Cyclic Redundancy Check, CRC). Циклічні надлишкові коди також називають поліноміальними кодами, тому що при їхньому обчисленні бітовий рядок розглядається як багаточлен (поліном), коефіцієнти якого рівні 0 або 1, і операції із цим бітовим рядком можна інтерпретувати як операції ділення і множення багаточленів.

Циклічні коди працюють у такий спосіб. Розглянемо фрагмент даних, що складається з d розрядів. Відправник і одержувач повинні домовитися про послідовність із $r+1$ біт, яка називається породжуючим багаточленом (або генератором). Старший (самий лівий) біт породжуючого багаточлена G повинен бути рівний 1. Ключову ідею циклічних надлишкових кодів ілюструє рис. 3.18.

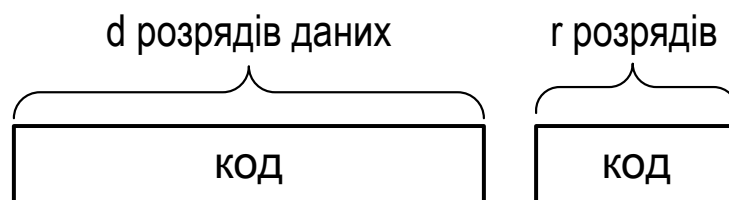


Рис. 3.18. Дані з CRC-кодом

Для заданого фрагмента даних відправник формує r додаткових розрядів, які він додає до даних так, щоб число, що виходить у результаті (воно складається із $d+r$ біт), ділилося по модулю 2 на породжуючий багаточлен без залишку. Таким чином, процес перевірки даних на наявність помилки відносно простий. Одержувач ділить отримані $d+r$ біт на породжуючий багаточлен. Якщо залишок від ділення не дорівнює нулю, це означає, що дані пошкоджені і ним не можна довіряти. А якщо ні, то дані вважаються вірними і приймаються [65].

3.5. Питання та завдання для самоперевірки

1. Які основні функції покладаються на систему комп'ютерного керування мехатронного об'єкта?
2. Назвіть способи організації обчислювального пристрою системи комп'ютерного керування.
3. Приведіть приклади різних типів інтерфейсів системи комп'ютерного керування мехатронного об'єкта.
4. Що таке промислова мережа?
5. Які типи даних вам відомі у розподілених системах на основі промислових мереж?
6. Зобразіть правильну та неправильну топологію мережі на основі інтерфейса RS-485.
7. В яких випадках бажано використовувати безпроводну передачу даних у мехатронних системах?
8. Вкажіть проблеми, які притаманні безпроводним мережам та шляхи їх вирішення.
9. Які переваги та недоліки характерні для безпроводної мережі Bluetooth?
10. Яка максимальна швидкість передачі даних у безпроводній мережі Wi-Fi?
11. Назвіть методи перевірки помилок у даних, які передаються каналом зв'язку, та поясніть їх суть.

4. ПРОЕКТУВАННЯ ТА ЕКСПЛУАТАЦІЯ МЕХАТРОННИХ ПРИБОРІВ

4.1. Проектування мехатронних систем і пристроїв

4.1.1 Системний підхід до проектування

Проектування мехатронних систем і пристроїв базується на використанні інформаційних технологій, зокрема, технології автоматизації проектування. Проблему підвищення конкурентоспроможності наукомістких товарів (до яких належать мехатронні системи й пристрої) можна вирішити за рахунок скорочення часу і матеріальних витрат виробництва й підвищення ефективності процесів життєвого циклу виробів, а також використання сучасних комп'ютерних систем на всіх етапах життєвого циклу виробів. У зв'язку з ускладненням виробів різко збільшується об'єм даних про виріб, і старі методи його проектування, виробництва й експлуатації вже не в змозі забезпечити необхідну точність, цілісність і актуальність. При великій кількості учасників проекту й виробництва виникають серйозні проблеми, пов'язані з обміном даними між учасниками, що виражаються в наявності комунікаційних бар'єрів.

Основний загальний принцип системного підходу полягає в розгляді частин явища або складної системи з врахуванням їх взаємодії. Системний підхід містить у собі виявлення структури системи, типізацію зв'язків, визначення атрибутів (властивостей), аналіз впливу зовнішнього середовища.

Інтерпретація й конкретизація системного підходу мають місце в ряді відомих підходів з іншими назвами: структурний, блочно-ієрархічний, об'єктно-орієнтований.

У системному підході введений ряд термінів, серед них до базових потрібно віднести приведені нижче поняття.

Система – множина елементів, що мають певні зв'язки між собою.

Елемент – така частина системи, уявлення про яку недоцільно піддавати подальшому розділенню при проектуванні.

Складна система – система, яка характеризується великим числом елементів і, що найбільш важливо, великим числом взаємозв'язків між елементами. Складність системи визначається також видом взаємозв'язків елементів, властивостями цілеспрямованості, цілісності, ієрархічності, багатоаспектності. Очевидно, що сучасні автоматизовані інформаційні системи й, зокрема, САПР є складними в силу наявності в них перерахованих властивостей і ознак.

Підсистема – частина системи (підмножина елементів і їх взаємозв'язків), яка має властивості системи.

Надсистема – система, стосовно якої розглянута система є підсистемою.

Структура – відображення сукупності елементів системи і їх взаємозв'язків; поняття структури відрізняється від поняття системи також тим, що при описі структури беруться до уваги лише типи елементів і зв'язків без конкретизації значень їх параметрів.

Параметр – величина, що виражає властивість системи, її частини або середовища, що впливає на систему. Звичайно в моделях систем у якості параметрів розглядають величини, що не змінюються в процесі дослідження системи. Параметри розділяють на зовнішні, внутрішні й вихідні властивості, що виражають властивості елементів системи, самої системи та зовнішнього середовища відповідно. Вектори внутрішніх, вихідних і зовнішніх параметрів позначаються $X = (x_1, x_2, \dots, x_n)$, $Y = (y_1, y_2, \dots, y_m)$, $Q = (q_1, q_2, \dots, q_k)$ відповідно.

Фазова змінна – величина, що характеризує енергетичне або інформаційне наповнення елемента або підсистеми.

Стан – сукупність значень фазових змінних, зафіксованих в деякий момент часу процесу функціонування.

Поведінка (динаміка) системи – зміна стану системи в процесі функціонування.

Система без післядії – її поведінка при $t > t_0$ визначається заданням стану в момент t_0 і вектором зовнішніх впливів $Q(t)$. У системах з післядією, крім того, потрібно знати передісторію поведінки, тобто стан системи в моменти часу, що передують t_0 .

Вектор змінних стану V – множина фазових змінних, задання значень яких у деякий момент часу повністю визначається поведінкою системи.

Простір стану – множина можливих значень вектора змінних стану.

Фазова траєкторія – представлення процесу (залежність $V(t)$) у вигляді послідовності точок у просторі стану.

До характеристик складних систем, як сказано вище, часто відносять наступні поняття.

Цілеспрямованість – властивість штучної системи, що виражає призначення системи. Ця властивість необхідна для оцінки ефективності варіантів системи.

Цілісність – властивість системи, що характеризує взаємозв'язок елементів і наявність залежності вихідних параметрів від параметрів елементів, при цьому більшість вихідних параметрів не є простим повторенням або сумою параметрів елементів.

Ієрархічність – властивість складної системи, що виражає можливість і доцільність її ієрархічного опису, тобто представлення у вигляді декількох рівнів, між компонентами яких є відношення „ціле – частина”.

Складовими частинами системотехніки є наступні основні розділи:

- ▶ ієрархічна структура систем, організація їх проектування;
- ▶ аналіз і моделювання систем;
- ▶ синтез і оптимізація систем.

Моделювання має два завдання: 1) створення моделей складних систем (в англійській літературі використовується термін – *modeling*); 2) аналіз

властивостей систем (в англomовній літературі використовується термін – *simulation*).

Синтез також розділяють на два завдання: 1) синтез структури проєктованих систем (структурний синтез); 2) вибір чисельних значень параметрів елементів систем (параметричний синтез). Ці завдання відносяться до області прийняття проєктних рішень. Моделювання й оптимізацію бажано виконувати з урахуванням статистичної природи системи. Детермінованість – лише окремий випадок. При проєктуванні характерні нестача достовірних вихідних даних, невизначеність умов прийняття рішень. Врахування статистичного характеру даних при моделюванні значною мірою засноване на методі статистичних випробувань (метод Монте-Карло), а прийняття рішень – на використанні нечітких множин, експертних систем, еволюційних обчислень.

Приклад 1. Комп'ютер є складною системою в силу наявності в нього великої кількості елементів, різноманітних зв'язків між елементами й підсистемами, властивостей цілеспрямованості, цілісності, ієрархічності. До підсистем комп'ютера відносяться процесор (процесори), оперативна пам'ять, кеш-пам'ять, шини, пристрої вводу-виводу. У якості надсистеми можуть виступати обчислювальна мережа, автоматизована й (або) організаційна система, до яких належить комп'ютер. Внутрішні параметри – тривалості виконання арифметичних операцій, зчитування (запис) у накопичувачах інформації, пропускну здатність шин тощо. Вихідні параметри – продуктивність комп'ютера, об'єм оперативної й зовнішньої пам'яті, собівартість, час наробітку на відмову тощо. Зовнішні параметри – напруга живлення мережі і його стабільність, температура навколишнього середовища тощо.

Приклад 2. Для двигуна внутрішнього згорання підсистемами є колінчатий вал, механізм газорозподілу, поршнева група, системи мащення й охолодження. Внутрішні параметри – число циліндрів, об'єм камери згорання тощо. Вихідні параметри – потужність двигуна, ККД, витрата

палива тощо. Зовнішні параметри – характеристики палива, температура повітря, навантаження на вихідному валу [66].

4.1.2. Структура процесу проектування

При використанні блочно-ієрархічного підходу до проектування представлення про проектувану систему розчленовують на ієрархічні рівні.

На верхньому рівні використовують найменш деталізоване представлення, що відбиває тільки самі загальні риси й особливості проектуваної системи. На наступних рівнях ступінь деталізації опису зростає, при цьому розглядаються вже окремі блоки системи, але з урахуванням впливів на кожний з них інших – „сусідніх” блоків. Такий підхід дозволяє на кожному ієрархічному рівні формулювати завдання прийнятної складності, що піддаються розв’язку за допомогою наявних засобів проектування. Розбивка на рівні повинна бути такою, щоб документація на блок будь-якого рівня було доступна для огляду й сприйманню однією людиною. Інакше кажучи, блочно-ієрархічний підхід є декомпозиційним підходом, який заснований на розбивці складного завдання великої розмірності на послідовно й (або) паралельно розв’язувані групи завдань малої розмірності, що суттєво скорочує вимоги до використовуваних обчислювальних ресурсів або час розв’язування завдань.

Список ієрархічних рівнів у кожному випадку може бути специфічним, але для більшості випадків характерне наступне виділення рівнів:

- системний рівень, на якому вирішуються найбільш загальні завдання проектування систем, машин і процесів; результати проектування представляють у вигляді структурних схем, генеральних планів, схем розміщення обладнання, діаграм потоків даних тощо;
- макрорівень, на якому проектують окремі пристрої, вузли машин і приладів; результати представляють у вигляді функціональних, принципових і кінематичних схем, складальних креслень тощо;

- мікрорівень, на якому проектують окремі деталі й елементи машин і приладів.

У кожному окремому випадку число рівнів, що виділяються, і їх найменування можуть бути різними. Так, у радіоелектроніці мікрорівень часто називають компонентним, макрорівень – схемотехнічним. В обчислювальній техніці системний рівень розділяють на рівні проектування ЕОМ (обчислювальних систем) і обчислювальних мереж. У машинобудуванні є рівні деталей, вузлів, машин, комплексів машин.

Залежно від послідовності розв’язування завдань ієрархічних рівнів розрізняють спадне, висхідне і змішане проектування (стилі проектування). Послідовність розв’язування завдань від нижніх рівнів до верхніх характеризує висхідне проектування, зворотна послідовність означає спадне проектування, у змішаному стилі є елементи як висхідного, так і спадного проектування. У більшості випадків для складних систем віддають перевагу спадному проектуванню. Відзначимо, однак, що при наявності заздалегідь спроектованих складених блоків (пристроїв) можна говорити про змішане проектування. Невизначеність і нечіткість вихідних даних при спадному проектуванні (оскільки ще не спроектовані компоненти) або вихідних вимог при висхідному проектуванні (оскільки технічне завдання дається на всю систему, а не на її частини) обумовлюють необхідність прогнозування відсутніх даних з наступним їхнім уточненням, тобто послідовного наближення до остаточного розв’язку (ітераційність проектування).

Поряд з декомпозицією описів на ієрархічні рівні застосовують поділ представлення про спроектовані об’єкти на аспекти. Аспект опису – опис системи або її частини з деякої застережливої точки зору, обумовленої функціональними, фізичними або іншого типу зв’язками між властивостями й елементами.

Розрізняють функціональний, інформаційний, структурний і поведінковий аспекти. Функціональний опис відносять до функцій системи й найчастіше представляють його функціональними схемами. Інформаційний

опис містить у собі основні поняття предметної області (сутності), словесне пояснення або числові значення характеристик (атрибутів) використовуваних об'єктів, а також опис зв'язків між цими поняттями й характеристиками. Інформаційні моделі можна представляти графічно (графи, діаграми „сутність – відношення”), у вигляді таблиць або списків. Структурний опис відноситься до морфології системи, характеризує складові частини системи і їх з'єднання й може бути презентований структурними схемами, а також різного роду конструкторською документацією. Поведінковий опис характеризує процеси функціонування (алгоритми) системи й (або) технологічні процеси створення системи. Іноді аспекти описів зв'язують із підсистемами, функціонування яких засновано на різних фізичних процесах. Відзначимо, що в загальному випадку виділення аспектів може бути неоднозначним. Так, крім зазначеного підходу очевидна доцільність виділення таких аспектів: функціонального (розробка принципів дії, структурних, функціональних, принципових схем), конструкторського (визначення форм і просторового розташування компонентів виробів), алгоритмічного (розробка алгоритмів і програмного забезпечення) і технологічного (розробка технологічних процесів) проектування систем.

4.1.3. Стадії проектування

Стадії проектування – найбільш великі частини проектування процесу, що розвивається в часі. У загальному випадку виділяють стадії науково-дослідних робіт (НДР), ескізного проекту або дослідно-конструкторських робіт, технічного, робочого проектів, випробувань дослідних зразків або дослідних партій. Стадію НДР іноді називають передпроектними дослідженнями або стадією технічної пропозиції. Очевидно, що в міру переходу від стадії до стадій ступінь деталізації й старанність проробки проекту зростають і робочий проект повинен бути цілком достатнім для виготовлення дослідних або серійних зразків.

Стадії (етапи) проектування розділяють на складові частини, які називаються проектними процедурами. Прикладами проектних процедур можуть служити підготовка креслень із деталюванням, аналіз кінематики, моделювання перехідного процесу, оптимізація параметрів і інші проектні завдання. У свою чергу, проектні процедури можна розчленувати на більш дрібні компоненти, які називаються проектними операціями, наприклад, при аналізі міцності деталі сітковими методами операціями можуть бути побудова сітки, вибір або розрахунки зовнішніх впливів, власне моделювання полів напруг і деформацій, представлення результатів моделювання в графічній і текстовій формах. Проектування зводиться до виконання деяких послідовностей проектних процедур – маршрутів проектування.

Іноді розробку технічного завдання на проектування називають зовнішнім проектуванням, а його реалізацію – внутрішнім проектуванням. У ТЗ на проектування об'єкта вказують наступні дані:

1. призначення об'єкта;
2. умови експлуатації. Поряд з якісними характеристиками (представленими у вербальній формі) є числові параметри, які називаються зовнішніми параметрами, для яких зазначені області допустимих значень. Приклади зовнішніх параметрів: температура навколишнього середовища, зовнішні сили, електричні напруги, навантаження тощо;
3. вимоги до вихідних параметрів, тобто до величин, що характеризують властивості об'єкта і які цікавлять споживача. Ці вимоги виражені у вигляді умов працездатності. Приклади умов працездатності: витрата палива на 100 км пробігу автомобіля менше 8л; коефіцієнт підсилення підсилювача на середніх частотах більше 300; швидкодія процесора більше 40 Мфлопс.

Створити проект об'єкта (виробу або процесу) означає вибрати структуру об'єкта, визначити значення всіх його параметрів і представити

результати у встановленій формі. Результати (проектна документація) можуть бути виражені у вигляді креслень, схем, пояснювальних записок, програм для програмно-керованого технологічного обладнання й інших документів на папері або машинних носіях інформації. Розробка (або вибір) структури об'єкта – проектна процедура, яка називається структурним синтезом, а розрахунки (або вибір) значень параметрів елементів X – процедура параметричного синтезу.

Завдання структурного синтезу формулюється в системотехніці як завдання прийняття рішень (ЗПР). Її суть полягає у визначенні мети, множини можливих розв'язків і обмежуючих умов. Класифікацію ЗПР здійснюють за рядом ознак. За числом критеріїв розрізняють завдання одно- і багатокритеріальні. За ступенем невизначеності розрізняють ЗПР детерміновані, ЗПР в умовах ризику (при наявності у формулюванні завдання випадкових параметрів), ЗПР в умовах невизначеності, тобто при неповноті або невірогідності вихідної інформації. Реальні завдання проектування, як правило, є багатокритеріальними.

Одна з основних проблем постановки багатокритеріальних завдань – встановлення правил переваги варіантів. Способи зведення багатокритеріальних завдань до однокритеріальних і наступні шляхи їх розв'язання вивчаються в дисциплінах, присвячених методам оптимізації. Наявність випадкових факторів ускладнює розв'язок ЗПР. Основні підходи до розв'язку ЗПР в умовах ризику полягають або в розв'язку „для найгіршого випадку”, або у врахуванні в цільовій функції математичного очікування й дисперсії вихідних параметрів. У першому випадку задачу розв'язують як детерміновану при завищених вимогах до якості розв'язку, що є головним недоліком підходу. У другому випадку вірогідність результатів розв'язку набагато вища, але виникають труднощі з оцінкою цільової функції. Застосування методу Монте-Карло у випадку алгоритмічних моделей стає єдиною альтернативою, і, отже, для розв'язку потрібні значні обчислювальні ресурси. Існують дві групи ЗПР в умовах невизначеності. Одна з них

вирішується при наявності протидії розумного супротивника. Такі завдання вивчаються в теорії ігор, для завдань проектування в техніці вони не характерні. У другій групі протидію досягненню мети виявляють сили природи. Для їхнього розв'язку корисно використовувати теорію й методи нечітких множин.

Наприклад, при синтезі структури автоматизованої системи постановка завдання повинна включати в якості вихідних даних наступні відомості:

- ▶ множину функцій, що виконуються системою (інакше кажучи, множину робіт, кожна з яких може складатися з однієї або більшої кількості операцій);
- ▶ типи допустимих для використання серверів (машин), що виконують функції системи;
- ▶ множину зовнішніх джерел і споживачів інформації;
- ▶ у багатьох випадках задається також деяка вихідна структура системи у вигляді взаємозалежної сукупності серверів певних типів, ця структура може розглядатися як узагальнена надлишкова або як варіант першого наближення;
- ▶ різного роду обмеження, зокрема обмеження на витрати матеріальних ресурсів і (або) на тривалість виконання функцій системи.

Завдання полягає в синтезі (або корекції) структури, визначенні типів серверів (програмно-апаратних засобів), розподілі функцій по серверах таким чином, щоб досягався екстремум цільової функції при виконанні заданих обмежень.

Конструювання, розробка технологічних процесів, оформлення проектної документації – окремі випадки структурного синтезу. Завдання параметричного синтезу називають параметричною оптимізацією (або оптимізацією), якщо її розв'язують як задачу математичного програмування, тобто:

$$F(X) \rightarrow \text{extr}, X \in D_X, \quad (4.1)$$

де $F(X)$ – цільова функція; X – вектор керованих (проектних) параметрів; D_x – допустима область ($D_x = \{X | \varphi(X) < 0, \Psi(X) = 0\}$), $\varphi(X)$ та $\Psi(X)$ – функції обмеження.

Приклад. Електронний підсилювач: керовані параметри X – параметри резисторів, конденсаторів, транзисторів; вихідні параметри Y – верхня й нижня граничні частоти смуги пропускання, коефіцієнт підсилення на середніх частотах, вхідний опір. У якості цільової функції $F(X)$ можна вибрати параметр верхньої граничної частоти смуги пропускання, а умови працездатності інших вихідних параметрів віднести до функцій-обмежень.

Наступна після синтезу група проектних процедур – процедури аналізу. Мета аналізу – одержання інформації про характер функціонування й значення вихідних параметрів Y при заданих структурі об'єкта, відомостях про зовнішні параметри Q і параметрах елементів X .

Якщо задані фіксовані значення параметрів X і Q , то має місце процедура одноваріантного аналізу, яка зводиться до розв'язку рівнянь математичної моделі й обчисленню вектора вихідних параметрів Y . Якщо задані статистичні відомості про параметри X і потрібно одержати оцінки числових характеристик розподілів вихідних параметрів (наприклад, оцінки математичних очікувань і дисперсій), то це процедура статистичного аналізу.

4.1.4. Концепція проектування мехатронних модулів і систем

Концептуальною основою проектування мехатронних систем є наступні положення [67]:

1. Основою розробки мехатронних систем є методи сполученого (паралельного) проектування, що припускають сполучення в часі виконання деяких етапів проектування, які виконуються послідовно при традиційному проектуванні (наприклад, послідовно виконується розробка механічної, електронної, сенсорної й комп'ютерної частин системи з наступною розробкою інтерфейсних блоків).

2. Синергетичний характер мехатронних систем визначає, що складові частини системи не просто доповнюють один одного, але поєднуються таким чином, що утворена ними система має нові властивості. При цьому мехатронні системи, на відміну від традиційних, мають більший ступінь інтеграції. У результаті підвищується конструктивна компактність системи (аж до мініатюризації в мікромашинах), поліпшуються масогабаритні й динамічні характеристики машин, спрощуються кінематичні ланцюги. При цьому виникають додаткові можливості при проектуванні мехатронних систем, які щонайкраще відповідають вимогам замовника.
3. Інтегровані мехатронні елементи вибираються розробником вже на стадії проектування машини, а потім забезпечується необхідна інженерна й технологічна підтримка при виробництві й експлуатації машини. У цьому радикальна відмінність мехатронних машин від традиційних, коли найчастіше користувач був змушений самостійно поєднувати в систему різноманітні механічні, електронні й інформаційно-керуючі пристрої різних виробників. Саме тому багато складних комплексів (наприклад, деякі гнучкі виробничі системи у вітчизняному машинобудуванні) показали на практиці низьку надійність і невисоку техніко-економічну ефективність.
4. Проектування мехатронних систем базується на ідеях модульного конструювання, при якому з окремих багатофункціональних модулів (механічної, електронної, комп'ютерної частин системи) компонується гнучкі складні системи модульної архітектури.
5. У мехатронних системах спостерігається перерозподіл функціонального навантаження від апаратних (наприклад, електромеханічних) модулів до інформаційних (комп'ютерних) модулів. При цьому виключається багатоступінчасте перетворення енергії й інформації.

6. При проектуванні мехатронних систем широко застосовуються методи візуалізації динамічних процесів, що протікають у системах, аж до використання систем віртуальної реальності при проектуванні складних технічних систем.
7. Високий рівень інтелектуалізації мехатронних систем диктує необхідність проектування інтелектуальних систем керування (для усунення невизначеностей, що супроводжують функціонування складних динамічних систем). При цьому необхідний обґрунтований вибір інтелектуальних технологій при проектуванні багаторівневих ієрархічних систем керування.

Практичне втілення даних принципів вимагає залучення всіх сучасних методів і засобів проектування складних систем. Необхідно відзначити, що проектування є одним із самих відповідальних етапів життєвого циклу промислових виробів. На цьому етапі в значній мірі закладається успіх виконання (реалізації) усіх інших етапів життєвого циклу, оскільки на цьому етапі визначаються всі параметри системи, технологічність її виробництва (виготовлення), вимоги до організації виробництва, експлуатаційні якості й можливість утилізації.

4.1.5. Алгоритм проектування

Проектування мехатронних модулів засноване на спільному аналізі їх функціональної, структурної й конструктивної моделей [68]. Метою проектування мехатронного модуля є перетворення вихідних вимог у конструкторську реалізацію й відповідну документацію, по якій може бути виготовлена ця система, що задовольняє сформульованим показникам якості. Обмежимося розглядом тільки технічних показників, але при виборі структури й конструкції мехатронних модулів слід також враховувати економічні й експлуатаційні показники якості. Загальний алгоритм проектування мехатронного модуля представлений на рис. 4.1. Даний алгоритм передбачає три основні етапи проектування – це послідовно

виконувані функціональний, структурний і конструктивний аналіз і синтез мехатронних модулів. Застосовуючи процедури функціонально-структурного й структурно-конструктивного аналізу, розробник оцінює прийняті рішення, прагнучи добитися високого рівня синергетичної інтеграції елементів. Для використання методів автоматизованого проектування формують взаємозалежні функціональну (*F*-модель), структурну (*S*-модель) і конструктивну (*C*-модель) моделі мехатронного модуля.

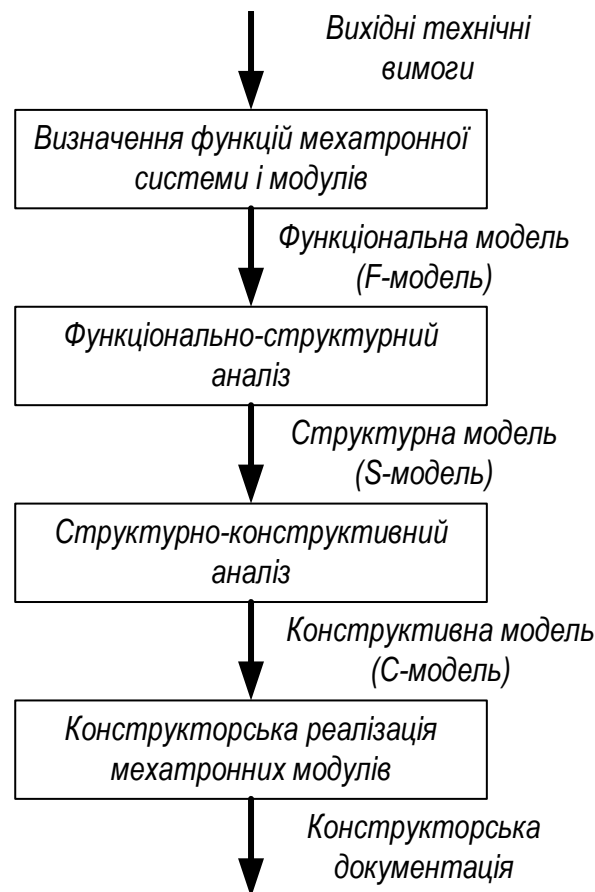


Рис. 4.1. Загальний алгоритм проектування мехатронного модуля

Визначення функції мехатронного модуля є проблемою концептуального проектування. Завдання проектування мехатронного модуля включає три основні етапи:

- 1) вибір варіантів структурних рішень модуля по його заданій функції і їх функціонально-структурний аналіз. Вхідною інформацією для цього етапу проектування є *F*-модель, а на виході формується *S*-модель мехатронного модуля;

- 2) структурно-конструктивний аналіз конструкторських рішень і побудова C-моделі мехатронного модуля;
- 3) конструкторська реалізація обраного варіанта модуля з розробкою конструкторської документації.

Таким чином, завдання проектування мехатронного модуля полягає в знаходженні найкращої відповідності між заданою функцією й конструктивним виконанням. Ключовою методологічною ідеєю даного підходу є пріоритет функції модуля над її структурною організацією й конструктивним рішенням. У мехатронному модулі задані функціональні перетворення можуть бути реалізовані декількома наборами структурних блоків, а ці блоки, у свою чергу, можуть мати різне конструкторське виконання. Таким чином, при проектуванні модуля його структура й конструктивна реалізація визначаються заданою функцією (вони є „підлеглими” до заданої функції).

Специфіка й складність мехатронних модулів полягає в тому, що їх складові частини (механічна, електронна й комп'ютерна) мають різну фізичну природу, а основні структурні елементи випускаються найчастіше підприємствами різних галузей промисловості [66].

4.1.6. Проектування мехатронної системи із мікропроцесорними пристроями

У загальному випадку процес проектування мехатронної системи із мікропроцесорними пристроями з погляду створення її програмного й алгоритмічного забезпечення можна розбити на сім основних етапів (рис. 4.2).

Етап 1. Аналіз технічного завдання. Змістом етапу є:

- визначення структурної побудови мехатронної системи;
- визначення функціональної побудови мехатронної системи;
- вибір елементної бази системи, у тому числі вибір мікроконтролера та його обв'язки;

- визначення переліку коригувальних пристроїв і законів керування, які передбачається реалізувати в мікроконтролері.

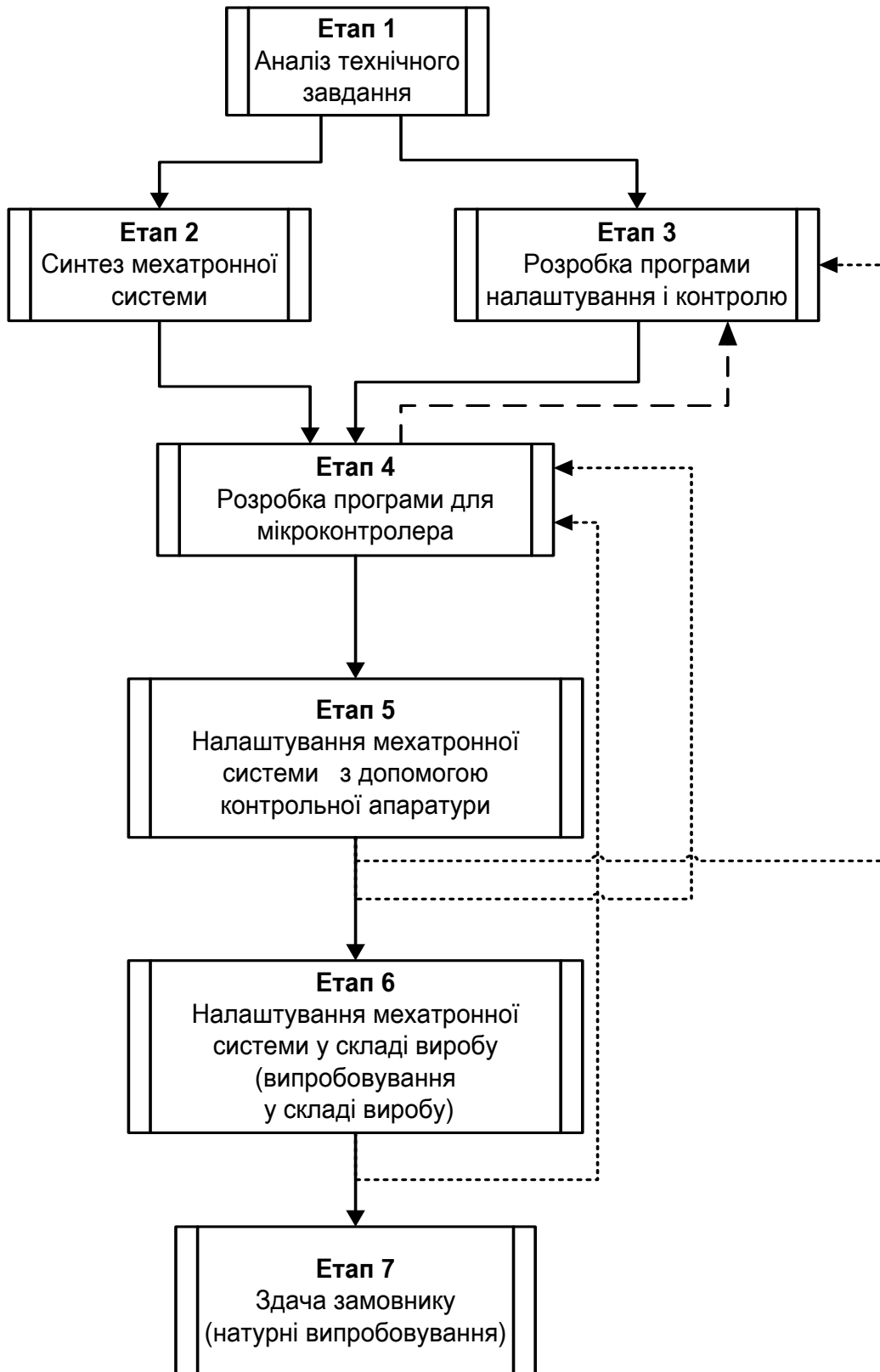


Рис. 4.2. Етапи проектування мехатронної системи мікропроцесорними пристроями

Етап 2. Синтез мехатронної системи (ескізний, технічний проекти).

Етап синтезу (у широкому сенсі) можна представити у вигляді п'яти кроків:

- енергетичний розрахунок силової частини мехатронної системи (остаточний вибір виконавчого пристрою, розглянутого на етапі аналізу технічного завдання);
- розробка математичної моделі незмінної частини мехатронної системи (розробка найпростішої математичної моделі для проведення „ручного” синтезу без врахування всіх особливостей, внесених мікропроцесорними пристроями керування);
- „ручний” синтез (синтез у вузькому сенсі) мехатронної системи (синтез лінійної системи традиційними „ручними” методами, наприклад, частотними, які бажано проводити із застосуванням програмного забезпечення для автоматизації розрахунків і побудови „точних” частотних характеристик. Із усіх особливостей, внесених мікропроцесором у динаміку системи, тут доцільно враховувати тільки квантування за часом. Цей крок закінчується визначенням передаточних функцій коригувальних пристроїв і законів керування виконавчим пристроєм, які є вихідною точкою для наступного моделювання);
- моделювання мехатронної системи (моделювання мехатронної системи із мікроконтролером за допомогою програмної моделі, яка реалізована на персональному комп'ютері);
- розробка блок-схеми алгоритмічного забезпечення мехатронної системи (блок-схема являє собою графічне відображення програмно-апаратної реалізації структурної побудови мехатронної системи у вигляді взаємодії джерел переривань і програм обробки цих переривань. Блок-схема служать вихідним матеріалом для розробки керуючої програми мікроконтролера).

Етап 3. Розробка програми налаштування й контролю мехатронної системи (технічний, робочий проекти). Метою розробки програми

налаштування й контролю мехатронної системи є створення засобу, що дозволяє автономно (поза мехатронним виробом) через персональний комп'ютер налагоджувати й контролювати роботу мехатронної системи із мікроконтролером, причому без „перепрошивання” постійного запам'ятовуючого пристрою мікроконтролера у випадку зміни параметрів коригувальних пристроїв, і по командах, що імітують роботу виробу.

Етап 4. Розробка керуючої програми мікроконтролера (робочий проект). Метою розробки керуючої програми мікроконтролера є програмна реалізація коригувальних пристроїв і законів керування мікроконтролером. Очевидно, що між керуючою програмою мікроконтролера і програмою налаштування й контролю повинна існувати однозначна відповідність.

Етап 5. Налаштування мехатронної системи із мікроконтролером за допомогою контрольної апаратури (автономні випробування). Метою етапу є попереднє налагодження програмно-апаратного забезпечення мехатронної системи. За результатами цього етапу можлива доробка програмного забезпечення мехатронної системи.

Етап 6. Налагодження мехатронної системи із мікроконтролером у складі виробу (випробування у складі виробу). Метою етапу є остаточне налагодження програмно-апаратного забезпечення мехатронної системи.

Етап 7. Здача замовникові (натурні випробування). Метою етапу є здача мехатронної системи замовникові.

Серед перерахованих вище етапів найбільший інтерес для проектування алгоритмічного й програмного забезпечення мехатронної системи представляють етапи 3 – 5 [69].

4.2. Мехатронні модулі - основа проектування сучасних мехатронних об'єктів

Відмінною рисою мехатронних об'єктів є синергетичне об'єднання в єдиному контурі систем різних типів - виконавчої (механічної), інформаційної (сенсорної) і керуючої (комп'ютерної). Усі ці системи мають

свої внутрішні „мови” представлення даних. Тому найважливішою частиною будь-якого мехатронного об’єкту є інтерфейси, що забезпечують „взаєморозуміння” окремих систем, що входять до складу мехатронного об’єкту.

Класична схема організації мехатронного об’єкту (рис. 4.3) включає до свого складу: систему комп’ютерного керування; приводи й механізми, що реалізують функції виконавчої системи; сенсори й функціональні перетворювачі, що реалізують функції інформаційної системи, а також інтерфейси чотирьох різних типів:

- цифро-аналоговий інтерфейс;
- електромеханічний інтерфейс;
- електричний інтерфейс;
- аналого-цифровий інтерфейс.

Прикладами механічних інтерфейсів є передачі та трансмісії, які пов’язують вихідний механічний пристрій з електродвигуном.

У цілому така схема мехатронного об’єкту працює в такий спосіб. За допомогою сенсорів (наприклад, тиску, переміщення, прискорення тощо) надходять дані про поточний стан зовнішнього середовища й механізмів мехатронного об’єкту. Ці дані формуються, як правило у вигляді напруги й струмів тобто в електричному виді, і несуть безпосередню інформацію про характеристики, що знімаються. Для того щоб виділити до цих опосередкованих даних корисну інформацію про стан середовища й мехатронного об’єкту, використовуються функціональні перетворювачі. Тому електричні дані із сенсорів передаються на функціональні перетворювачі за допомогою електричного інтерфейсу. Після перетворення дані передаються далі в систему керування за допомогою аналого-цифрового інтерфейсу.

У результаті обробки даних в системі керування згідно із запрограмованим алгоритмом формуються команди керування виконавчими механізмами мехатронного об’єкту. Ці команди, представлені на виході

системи керування в цифровому виді, повинні спочатку бути перетворені в аналоговий вид, доступний приводам механізмів. Для цієї мети служить цифро-аналоговий інтерфейс. Далі приводи формують керуючі впливи безпосередньо на механізми мехатронного об'єкту за допомогою електромеханічного інтерфейсу.

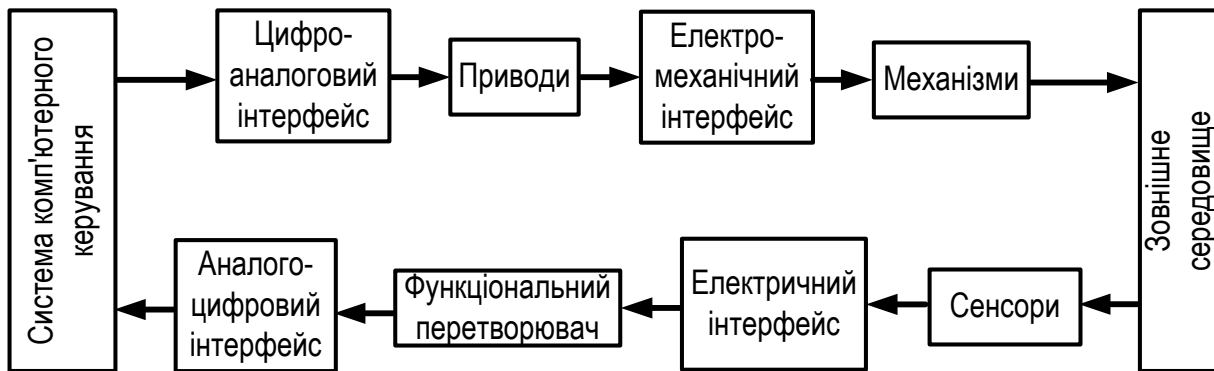


Рис. 4.3. Структурна схема мехатронного об'єкту

З рис. 4.3 видно, що проектування мехатронного об'єкту являє собою складний процес, що вимагає знань у таких областях як механіка, електромеханіка, електроніка, електрика й комп'ютерна техніка. На перших етапах розвитку мехатроніки проектування мехатронного об'єкту здійснювалося, як правило, групою розроблювачів, кожний з яких мав знання у своїй специфічній області, а їх взаємодія полягала в узгодженні інтерфейсів між окремими частинами мехатронного об'єкту. При цьому зрозуміло, що такий „слабозв'язаний” процес проектування не гарантував оптимальної структури мехатронного об'єкту й приводив до зниження його надійності внаслідок наявності „вузьких місць”, якими були інтерфейси між окремим складовими мехатронного об'єкту. Інакше кажучи, окремі частини мехатронного об'єкту, спроектовані різними розроблювачами, не завжди правильно „розуміли” один одного при обміні інформацією через інтерфейси.

Тому сучасні методи проектування мехатронного об'єкту використовують інші підходи, засновані на застосуванні так званих мехатронних модулів. Основна мета застосування мехатронних модулів - це

скорочення числа точок „зіткнення” складових частин мехатронного об’єкту (інтерфейсів) і відповідно, точок „спотворення” інформації.

Під мехатронним модулем розуміється закінчений пристрій, що володіє уніфікованим інтерфейсом і виконує ті або інші функції мехатронного об’єкта. Розрізняють мехатронні модулі двох типів: виконавчі модулі (модулі руху), що реалізують функції механічного руху мехатронного об’єкту й інформаційні модулі (датчики), що здійснюють функції інформаційного забезпечення мехатронного об’єкту.

Існують різні класифікації етапів еволюції мехатронних модулів. Нас більшою мірою буде цікавити еволюція мехатронних модулів з погляду включення до їхнього складу тих або інших інтерфейсів. Із цього погляду можна виділити два покоління виконавчих і інформаційних модулів. Виконавчий модуль першого покоління об’єднує у своєму складі привод й власне механізм за допомогою електромеханічного інтерфейсу (рис. 4.4).

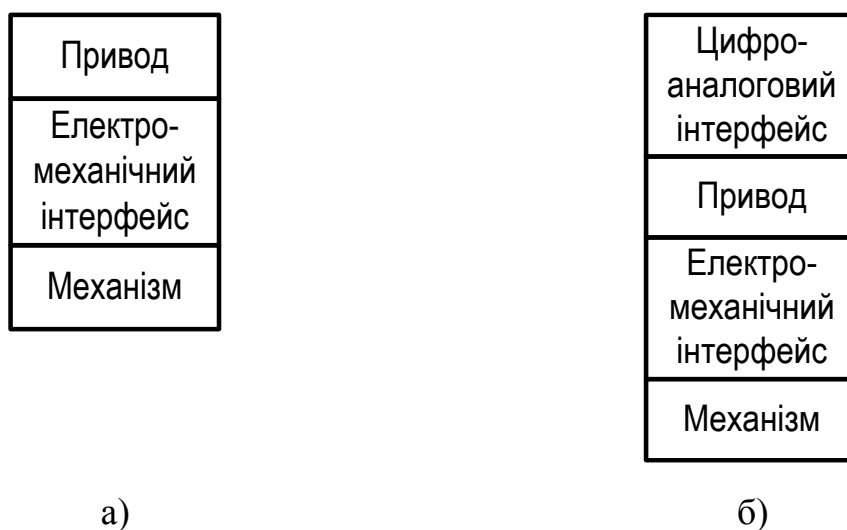


Рис. 4.4. Типи мехатронних виконавчих модулів: а) першого покоління; б) другого покоління

Такий мехатронний модуль ще називають приводним мехатронним модулем. Виконавчий мехатронний модуль другого покоління поєднує у своєму складі крім приводу, механізму й електромеханічного інтерфейсу також цифро-аналоговий інтерфейс, що забезпечує можливість прямого

підключення мехатронного модуля безпосередньо до системи керування (рис. 4.5).

Аналогічно інформаційний модуль першого покоління включає до свого складу сенсор, функціональний перетворювач і електричний інтерфейс (рис. 4.5, а), а інформаційний модуль другого покоління включає до свого складу також аналого-цифровий інтерфейс, що забезпечує можливість прямого підключення до системи керування (рис. 4.5, б).

Відзначимо, що деякі автори класифікують ще й мехатронні модулі третього покоління, включаючи до їхнього складу, крім інформаційної або виконавчої частини, процесорний вузол. Однак, уведення до складу мехатронного модуля процесора означає прирівнювання його до елементарного мехатронного об'єкта. Тому в подальших побудовах будемо використовувати дворівневу класифікацію мехатронних модулів.

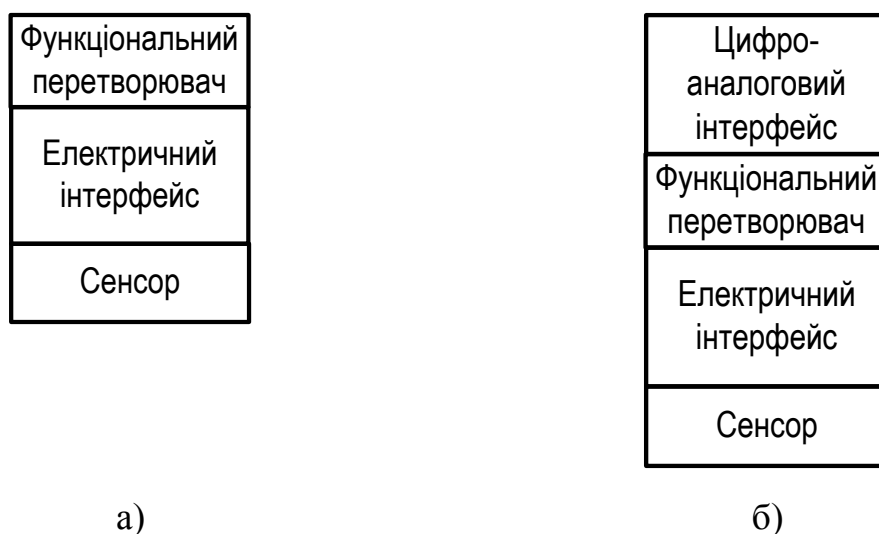


Рис. 4.5. Типи інформаційних модулів: а) першого покоління; б) другого покоління

Відмінність між процесами проектування мехатронних об'єктів з використанням мехатронних модулів 1-го й 2-го поколінь показана на рис. 4.6. Як ми вже відзначали вище, у випадку відсутності готових мехатронних модулів при проектуванні мехатронних об'єктів потрібна окрема розробка

механічної, електромеханічної, інформаційної й керуючої комп'ютерної частин, а також узгодження інтерфейсів зв'язки між ними (рис. 4.3).

Застосування мехатронних модулів першого покоління дозволяє виключити із процесу проектування процедури розробки й узгодження електромеханічних і електричних інтерфейсів тим самим спростити процес проектування й підвищити надійність експлуатації мехатронних об'єктів в цілому (рис. 4.5, а).

Використання мехатронних модулів другого покоління дозволяє в ще більшому ступені спростити структуру мехатронних об'єктів й відповідно, знизити складність і вартість його проектування й експлуатації (рис. 4.5, б). Підкреслимо, що використання мехатронних модулів другого покоління дозволить перейти до цифрових інтерфейсів зв'язку між окремими частинами мехатронних об'єктів, які в цей час знаходять усе більш широке застосування при створенні складних мехатронних об'єктів.

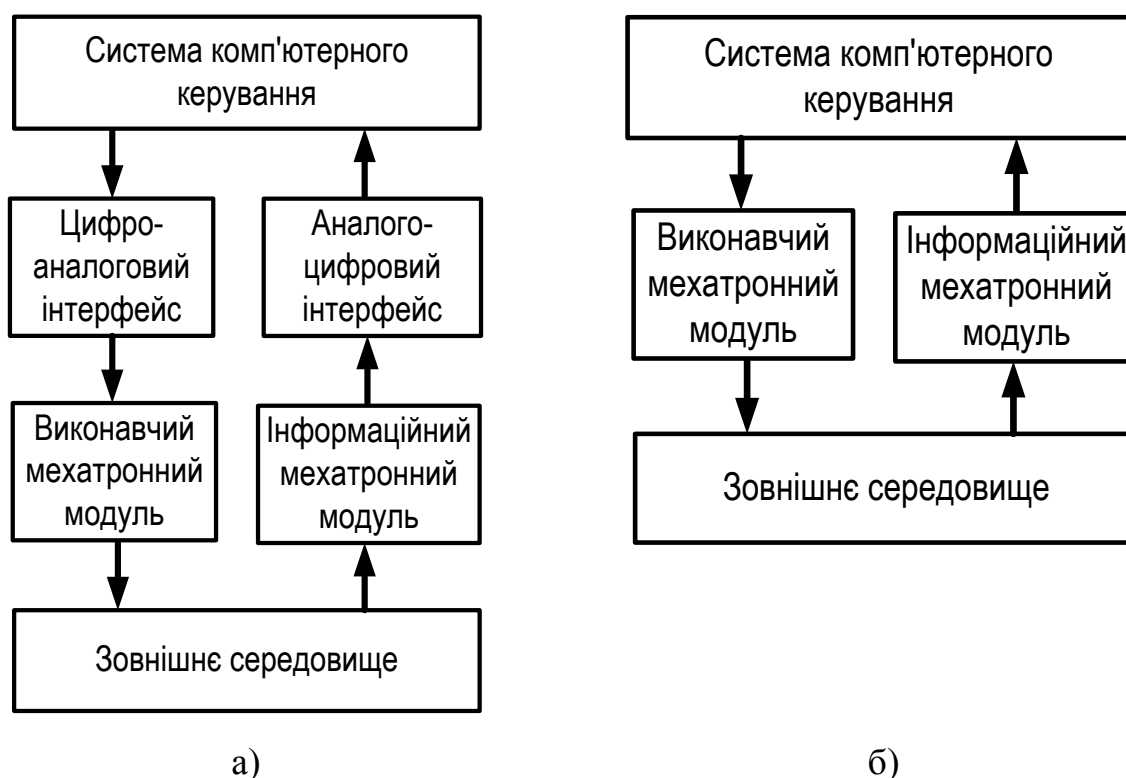


Рис. 4.6. Структура мехатронних об'єктів при використанні мехатронних модулів: а) першого покоління; б) другого покоління

Використання концепції мехатронних модулів дозволяє в значній мірі абстрагуватися в процесі проектування складного мехатронного об'єкту від виконавчої й інформаційних частин, вважаючи, що вони компонуються з готових „кубиків” - мехатронних модулів, а зосередити основну увагу на розробці системи комп'ютерного керування, що є „мозком” будь-якого мехатронного об'єкта. Саме від ефективності проектування систем керування мехатронного об'єкта найбільшою мірою буде залежати ефективність функціонування всього мехатронного об'єкту в цілому. Тому надалі основну увагу буде приділено саме проблемам розробки й проектування систем керування мехатронного об'єкта [52].

4.3. Розробка програмної частини мехатронного пристрою

4.3.1. Рівні програмування

Машинно-орієнтоване програмування

З винаходом обчислювальних машин, які керуються за певними програмами, з'явилася нова професія - програміст. Першим в історії програмістом була Ада Лавлейс, яка працювала разом з Чарльзом Баббіджом, вона розробляла програми керування його Аналітичною машиною. Масовою професією програміста стала тільки з винаходом електронних обчислювальних машин - ЕОМ.

На лампових ЕОМ першого покоління програмісти складали свої програми, використовуючи безпосередньо команди процесора. При цьому програмістові доводилося розподіляти комірки пам'яті під дані й під команди програми. Потрібно було знати систему команд процесора й коди всіх команд. Вихідні дані й команди представлялися у формі двійкового коду, тобто безпосередньо в тому виді, у якому вони зберігалися в пам'яті ЕОМ. Для скорочення запису програм на спеціальних бланках звичайно використовували двійково-шіснадцятковий код. У таблиці 4.1 приведено приклад програми для одного із комп'ютерів першого покоління.

Таблиця. 4.1. Команда додавання у машинному кодi

	Адреса команди	Код операції	1-а адреса	2-а адреса	3-а адреса
Шіснадцятковий код	28	02	C0	C4	D8
Двійковий код	00101000	00000010	11000000	11000100	11011000

Команда, яка описується в табл. 4.1 називається триадресовою. Код 02_{16} відноситься до команди додавання. Перша та друга адреси – це адреси комірок оперативної пам'яті, в яких зберігаються доданки, третя адреса – адреса комірки куди заноситься сума. Сама команда зберігається у комірці оперативної пам'яті з адресою 28_{16} .

Програмування в машинних кодах являло собою складний процес. Саме тому продуктивність роботи програмістів була досить низкою. В 1950-х роках виникає напрямок, який одержав назву „автоматизація програмування”. Основна його мета - створення засобів, що полегшують і прискорюють процес створення програми для ЕОМ. З'являються перші мови програмування.

Першими мовами програмування були машинно-орієнтовані автокоди. Пізніше за мовами такого рівня закріпилася назва „асемблери”. Спочатку асемблером називали програму-перекладач із мови асемблера в машинні команди. Пізніше саму мову асемблера стали називати іменем „асемблер”. Програмування на асемблері дозволяє програмістові не розділяти пам'ять під дані й команди програми. Програміст також може не пам'ятати внутрішні коди всіх команд процесора. Приведемо приклад тієї ж команди додавання на асемблері:

ADD a, b, c

Слово ADD позначає команду „додати”, а й b — імена змінних-доданків, c - змінна, куди записується результат.

Мова асемблер називається машинно-орієнтованою з тієї причини, що для кожної команди процесора існує свій аналог команди на асемблері. Оскільки різні типи ЕОМ мали різні системи команд процесора, то асемблери в них різнилися. Сучасні асемблери аналогічно орієнтовані на певні типи процесорів. Пізніше з'явилися так звані макроасемблери, у мові яких існують макрокоманди, відповідні до серій команд (підпрограмам) мови процесора.

Складання програми на асемблері є простішим, ніж мовою команд процесора. Роботу з розподілу пам'яті під дані й команди, переклад команд асемблера в машинні команди бере на себе спеціальна системна програма - транслятор.

З машинної орієнтованості програм на асемблері випливає, що такі програми не можна переносити для виконання на інші типи ЕОМ з іншою системою команд процесора. Ця проблема створила серйозні обмеження для прикладних програмістів. Крім того, програмування на асемблері є досить складним для масового освоєння, що обмежило використання ЕОМ у прикладних областях.

Мови програмування високого рівня

Наступним етапом розвитку програмування стало створення мов програмування високого рівня (МПВР). Приклади МПВУ: Паскаль, Бейсік, Фортран. Для кожної мови існує машинно-незалежний стандарт. Можливість програмування на даній МПВР залежить від наявності на комп'ютері транслятора із цієї мови. Транслятори для кожного типу комп'ютерів складають системні програмісти.

Текст програми на МПВР за своєю формою ближче до природних мов (найчастіше до англійської) та до мови математики. Та ж команда додавання двох величин на МПВР схожа на звичну форму математичної рівності:

$c := a + b$ (на Паскалі);

$c = a + b$ (на Фортрані, Бейсіку, Сі).

Набагато простіше освоїти програмування МПВР, ніж програмування на асемблері. Тому з появою МПВР значно зросло число прикладних програмістів, розширилося застосування ЕОМ у багатьох областях.

Починаючи із середини ХХ століття й до нашого часу були створені сотні мов програмування високого рівня. Але розповсюдженими й популярними з них стали не всі. Одним з довгожителів у сімействі МПВР є мова Фортран. Перша версія Фортрану була створена в 1954 році. У часи ЕОМ другого й третього поколінь була популярна версія Фортран-IV. Фортран створювалася як спеціалізована мова для математичних розрахунків, використовуваних у науці й техніці. І в наш час ця мова, у стандарті Фортран-90 (і в наступних його модифікаціях Фортран-95, Фортран-2003), залишається важливою засобом програмування для розрахунків в області фізико-технічних проблем.

Велика кількість мов програмування з'явилася в 1960- 1970-х роках. В 1965 році в Дартмутському університеті була розроблена мова Бейсік. За задумом авторів це повинна бути проста мова, яка легко вивчається і призначена для програмування нескладних обчислювальних завдань. Найбільше поширення Бейсік одержав з появою мікро-ЕВМ і персональних комп'ютерів.

Значною подією в історії мов програмування стало створення в 1969 році мови Паскаль. Його автор - швейцарський професор Ніклаус Вірт розробив Паскаль як навчальну мову структурного програмування.

Найбільший успіх у поширенні мови Паскаль забезпечили персональні комп'ютери. Фірма Borland International, Inc (США) розробила систему програмування Turbo Pascal (Турбо Паскаль) для ПК. Турбо Паскаль - це не тільки мова та транслятор з неї, але ще й інтегроване середовище програмування, що дає користувачеві можливість зручно працювати на Паскалі: вводити й редагувати текст програми, шукати синтаксичні помилки, користуватися бібліотеками підпрограм і модулів, працювати з файлами тощо. Турбо Паскаль вийшов за рамки навчального призначення й став

мовою професійного програмування з універсальними можливостями. Паскаль став джерелом багатьох основних сучасних мов програмування, наприклад таких, як Ада, Модула-2 та інші.

Модула-2 - це ще одна мова, запропонована Віртом, яка є розвитком мови Паскаль. Вона включає засоби для створення великих програм.

Мова програмування Сі (англійська назва - С) з'явився практично одночасно з Паскалем. Вона створювалася як інструментальна мова для розробки операційних систем, трансляторів, баз даних і інших системних і прикладних програм. Хоча Сі і є мовою високого рівня, однак у ньому закладені можливості безпосереднього звертання до деяких машинних команд, до певних ділянок пам'яті комп'ютера, що раніше було можливе тільки в асемблері. З появою Сі багато системних програмістів перейшли з асемблера на Сі. Подальший розвиток Сі привів до створення мови об'єктно-орієнтованого програмування Сі++ [70].

4.3.2. Парадигми програмування

Слово „парадигма” стосовно до програмування означає певний загальноприйнятий підхід до організації обчислень на комп'ютері. Парадигма визначає систему базових понять, на основі якої відбувається програмування.

Мови програмування, про яких йшлося вище, засновані на **процедурній парадигмі програмування**. Алгоритм, реалізований на процедурній мові, ґрунтується на уявленнях про фон-неймановську архітектуру комп'ютера. Базовим поняттям є поняття величини, яка зберігається в пам'яті комп'ютера, базовою операцією - операція присвоювання. Розв'язок завдання на комп'ютері відбувається шляхом зміни стану пам'яті: на початку в пам'ять записуються вихідні дані, наприкінці отримують результати. Програма - це опис процедури переходу пам'яті з початкового стану в кінцевий в процесі виконання послідовності команд. Мислення програміста, що працює в

рамках процедурної парадигми, спрямоване на зведення розв'язку завдання до послідовності команд, які вмiє виконувати процесор комп'ютера.

На іншому підході до програмування заснована **функціональна парадигма програмування**. Найбільш відомою мовою, що реалізують функціональну парадигму, є ЛІСП. Перше згадування про ЛІСП відноситься до 1958 року. Ця мова створена на основі поняття рекурсивної функції. У теоретичному програмуванні доведено, що будь-який алгоритм може бути описаний за допомогою деякого набору рекурсивних функцій. Тому ЛІСП, по суті, є універсальною мовою. З його допомогою на комп'ютері можна моделювати досить складні процеси, зокрема, інтелектуальну діяльність людей. Крім ЛІСП існують і інші мови функціонального програмування: Haskell, Scheme та інші.

Логічна парадигма програмування реалізована в мові Пролог. Пролог був розроблений у Франції в 1972 році. Його, як і ЛІСП, відносять до мов штучного інтелекту. Пролог заснований на апараті математичної логіки, звідси назва парадигми. На програмі Пролог будується база знань у певній предметній області, що полягає із сукупності фактів і правил. Для розв'язання деякого завдання формулюється запит до бази знань, який називається метою. Реалізація мети (відповідь на запит) відбувається через закладений в інтерпретатор програми Пролог алгоритм, який називається механізмом виводу. Використовуючи Пролог, можна вирішувати логічні завдання. Слід зазначити, що за допомогою програми Пролог можна вирішувати лише обмежене коло завдань штучного інтелекту. До мов логічного програмування також відносяться: Planner, Mercury, Fril та інші.

Об'єктно-орієнтована парадигма програмування заснована на концепції об'єктів і класів. Всякий об'єкт характеризується набором властивостей і дій, які з ним можуть бути виконані і які він може виконувати. Об'єкт відноситься до певного класу. Об'єктно-орієнтоване програмування зводиться до вибудовування ієрархій класів, описів об'єктів і їх взаємодій, програмної реалізації різних дій над об'єктами. Першою мовою даної

парадигми програмування була Симула, розроблена в 1967 році. Пізніше елементи об'єктно-орієнтованого програмування стали впроваджуватися в процедурні мови, у тому числі в Турбо Паскаль. Найбільш популярними засобами об'єктно-орієнтованого програмування є мови Сі++, Java [70].

4.3.3. Методології й технології програмування

На перших ЕОМ з „тісною” пам'яттю й невеликою швидкістю основним показником якості програми була її економічність по займаній пам'яті й часу розрахунку.

З ростом пам'яті й швидкодії ЕОМ, з удосконалюванням мов програмування й трансляторів із цих мов проблема економічності програми стає менш гострою. Все більш важливою якісною характеристикою програм стає їхня простота, наочність, надійність. З появою машин третього покоління ці якості стали основними.

Уже в 1960-х роках програмування стало досить масовою професійною діяльністю. Виникають компанії (фірми) з розробки програм. Актуальним стає завдання розробки загальноприйнятої методології програмування, що підвищує продуктивність роботи програмістів і, що найголовніше, якість програмних продуктів. Основний якісний показник програми - її працездатність, відсутність помилок.

Методологія програмування - це сукупність певних способів написання, налагодження й супроводу програм. Перша найбільш відома й розповсюджена методологія програмування одержала назву „структурне програмування”.

Поява структурного програмування пов'язана з іменами Едсгера Дейкстри та Чарльза Хоара. Починаючи з 1960-х років стали з'являтися мови структурного програмування. Першим з них був Алгол-60, розроблений Дейкстрою, потім був створений Паскаль. Інші, спочатку „не структурні” мови стали також здобувати „структурні властивості” (Турбо Бейсік, Фортран-77 та інші). Структурне програмування - методологія розробки

програмного забезпечення, в основі якої лежить представлення програми у вигляді ієрархічної структури блоків. Відповідно до даної методології:

1. Будь-яка програма являє собою структуру, побудовану із трьох типів базових конструкцій:

- послідовне виконання - однократне виконання операцій у тому порядку, у якому вони записані в тексті програми;
- розгалуження - однократне виконання однієї із двох або більшої кількості операцій, залежно від виконання деякої заданої умови;
- цикл - багаторазове виконання однієї й тієї ж операції доти, поки виконується деяка задана умова (умова продовження циклу).

У програмі базові конструкції можуть бути вкладені один у одного довільно, але ніяких інших засобів керування послідовністю виконання операцій не передбачається.

2. Повторювані фрагменти програми або не повторювані, але такі, що представляють собою логічно цілісні обчислювальні блоки, можуть оформлятися у вигляді так званих підпрограм (процедур або функцій).

У цьому випадку в тексті основної програми, замість поміщеного в підпрограму фрагмента, вставляється інструкція виклику підпрограми.

При виконанні такої інструкції виконується викликана підпрограма, після чого виконання програми починається з інструкції, що йде за командою виклику підпрограми.

3. Розробка програми ведеться покроково, методом „зверху вниз”.

Спочатку пишеться текст основної програми, у якому замість кожного зв'язного логічного фрагмента тексту, вставляється виклик підпрограми, яка буде виконувати цей фрагмент. Замість справжніх працюючих підпрограм у програму вставляються „заглушки”, які нічого не роблять. Отримана програма перевіряється й налагоджується. Після того, як програміст переконується, що підпрограми викликаються в правильній послідовності (тобто загальна структура програми вірна), підпрограми-заглушки послідовно замінюються на реально працюючі, причому розробка кожної

підпрограми ведеться тим же методом, що й основної програми. Розробка закінчується тоді, коли не залишиться ні однієї „заглушки”, яка не була б замінена. Така послідовність гарантує, що на кожному етапі розробки програміст одночасно має справу з доступною й зрозумілою йому множиною фрагментів. Він може бути впевнений, що загальна структура всіх більш високих рівнів програми вірна. При супроводі й внесенні змін у програму з’ясовується, у які саме процедури потрібно внести зміни, і вони вносяться, не зачіпаючи частини програми, безпосередньо не пов’язаних з ними. Це дозволяє гарантувати, що при внесенні змін і виправленні помилок не вийде з ладу якась частина програми, що перебуває в цей момент поза зоною уваги програміста [71].

Структурне програмування дотепер залишається найважливішою методологією програмування. Дотримання його принципів дозволяє програмістові розробляти ясні, безпомилкові, надійні програми.

Технологія - це масовий спосіб виробництва якогось продукту, що гарантує стійкий результат. Технологія припускає наявність певних засобів, інструментів виробництва. В 1990-х роках з розвитком об’єктно-орієнтованої парадигми програмування, а також засобів графічного інтерфейсу на персональних комп’ютерах, виникає нова технологія програмування - візуальне програмування, що представляє собою спосіб створення програми для ЕОМ шляхом маніпулювання графічними об’єктами замість написання її коду. Візуальна технологія програмування дозволяє програмістові легко й швидко будувати наочний графічний інтерфейс для своїх програм на основі стандартного набору шаблонів, що графічно відображуються на екрані об’єктів [70].

4.3.4. Огляд програмних засобів, що використовуються при проектуванні програмного забезпечення мікроконтролерів

4.3.4.1. Програмні засоби для розробки програмного забезпечення мікроконтролерів

Основним засобом одержання кінцевого програмного продукту є компілятор. Під компілятором розуміється спеціальна програма, що переводить вихідні тексти конкретної мови програмування в модуль, що виконується, тобто у коди машинних команд даного процесора (мікроконтролера). Для розробки керуючої програми мікроконтролера у цей час використовуються практично тільки дві мови програмування – асемблер та Сі. Це визначається широким поширенням компіляторів тільки цих двох мов при програмуванні різних процесорів і мікроконтролерів. Що стосується інших мов, наприклад, Forth, то вона не отримала широкого поширення.

Асемблер

Асемблером прийнято називати машинно-орієнтовану мову, у якій кожній мнемонічній команді відповідає одна машинна інструкція (за винятком макрокоманд). Макроасемблером називається асемблер, у якому використовується механізм макропідстановок (макрокоманд). Суть макропідстановки полягає в заміні на першому кроці компіляції символічного імені макрокоманди відповідним йому блоком програмного коду. Перевагою асемблера над іншими мовами є мінімальний розмір модуля, що виконується, і значна швидкодія. Застосування асемблера вимагає глибокого знання архітектури й логіки мікроконтролера і висуває високі вимоги до програміста.

Сі

Мова Сі відноситься до мов високого (стосовно асемблера) рівня, але через свою здатність працювати з апаратурою мікроконтролера із такою ж ефективністю, як і асемблер, одержала назву „асемблер високого рівня”. Перевагою Сі над асемблером є зменшення трудомісткості розробки

керуючої програми при незначному зниженні її швидкодії. Однак використання Сі як і раніше вимагає глибокого знання архітектури й логіки мікроконтролера, а також особливостей модифікації мови Сі конкретного компілятора, специфічних для даного типу мікропроцесора або мікроконтролера [69].

4.3.4.2. Програмні засоби для налаштування і контролю програм мікроконтролерів

Мовами програмування, які доцільно використовувати для налаштування і контролю програм мікроконтролера, є Сі та Паскаль. У ряду обставин найбільше поширення одержала мова Сі. Програму налаштування й контролю можна розробити як у варіанті „під DOS”, так і у варіанті „під Windows”. У випадку зв’язку програми налаштування й контролю з мікроконтролером через асинхронні послідовні інтерфейси RS-232 або RS-485 обидва варіанти з погляду ефективності налаштування мехатронної системи із мікроконтролерами однакові. Варіант програми налаштування й контролю „під DOS” забезпечує програмістові повний доступ до всіх внутрішніх ресурсів персонального комп’ютера, включаючи реєстри послідовних портів COM, вектори різних переривань і програми з їхнього обслуговування. Варіант „під DOS” має текстовий інтерфейс (введення команд ведеться виключно із клавіатури), і потрібні певні зусилля по організації роботи з мишею.

Варіант „під Windows” має графічний інтерфейс, більш приємний для користувача, але не допускає програміста до внутрішніх ресурсів комп’ютера. У загальному випадку вибір варіанту реалізації програми налаштування й контролю є питанням особистих вподобань і традицій. Слід відмітити, що варіант „під DOS” може працювати в середовищі як „чистої” DOS, так і в різних операційних середовищах Windows, а варіант „під Windows” практично не придатний для „старих” комп’ютерів з обмеженим об’ємом пам’яті. Однак у будь-якому варіанті програма налаштування й

контролю повинна являти собою систему меню й віконних повідомлень, найбільш зручну для ефективної роботи.

Для розробки варіанта „під DOS” на Сі можуть використовуватися компілятори фірми Borland – TurboC, Borland C++, а також фірми Microsoft (Quickc 2). Для розробки варіанта „під DOS” на Паскалі можуть використовуватися компілятори фірми Borland – Turbo Pascal.

Для розробки варіанта „під Windows” на Сі можуть використовуватися інтегровані середовища програмування Microsoft VisualC, VisualC++, Borland C++Builder. Для розробки варіанта „під Windows” на Паскалі може використовуватися інтегроване середовище програмування Borland Delphi [69].

У додатку 2 приведено приклад розробки програмного забезпечення та апаратної частини системи керування електроприводом моделі вантажопідйомного крана.

4.4. Нанотехнології в мехатроніці

Нанотехнології є дуже перспективним, але поки не розвиненими повною мірою, напрямком досліджень. Нанотехнологія – міждисциплінарна область фундаментальної й прикладної науки й техніки, що має справу із сукупністю теоретичного обґрунтування, практичних методів дослідження, аналізу й синтезу, а також методів виробництва й застосування продуктів із заданою атомарною структурою шляхом контрольованого маніпулювання окремими атомами і молекулами.

Нанороботи, або наноботи – роботи, створені з наноматеріалів і розміром порівнянним із молекулою, що володіють функціями, обробки й передачі інформації, виконання програм. Нанороботи, які здатні до створення своїх копій, тобто мають функцію самовідтворення, називаються реплікаторами. У цей час уже створені електромеханічні нанопристрої, обмежено здатні до пересування, які можна вважати прототипами нанороботів.

4.4.1. Історія розвитку нанотехнології в датах

Один нанометр (від грецького „нано” – карлик) рівній одній мільярдній частині метра. На цій відстані можна впритул розташувати приблизно 10 атомів. Мабуть, перший ученим, що використовували цю одиницю виміру, був Альберт Ейнштейн, який в 1905 р. теоретично довів, що розмір молекули цукру рівний одному нанометру.

Але тільки через 26 років німецькі фізики Ернст Руска, що одержав Нобелівську премію в 1986 р., і Макс Кнолл створили електронний мікроскоп, що забезпечує 15-кратне збільшення (менше, чим існуючі тоді оптичні мікроскопи), він і став прообразом нового покоління подібних пристроїв, що дозволили зазирнути в наносвіт.

В 1932 р. голландський професор Фріц Церніке, Нобелівський лауреат 1953 р., винайшов фазово-контрастний мікроскоп – варіант оптичного мікроскопа, що поліпшував якість деталей зображення, і досліджував з його допомогою живі клітини (раніше для цього доводилося застосовувати барвники, що вбивали живі тканини). Цікаво, що Церніке пропонував свій винахід фірмі «Цейс», але менеджери не усвідомили його перспективності, хоча сьогодні такі мікроскопи активно застосовуються в медицині.

В 1939 р. компанія Siemens, у якій працював Руска, випустила перший комерційний електронний мікроскоп з роздільною здатністю 10 нм.

Днем народження нанотехнологій вважається 29 грудня 1959 р. Професор Каліфорнійського технологічного інституту Річард Фейнман (Нобелівський лауреат 1965 р.) у своїй лекції „Як багато місця там, знизу” („There’s plenty of room at the bottom”), яку він прочитав перед Американським фізичним об’єднанням, відзначив можливість використання атомів як будівельних часток.

1966 р. Американський фізик Рассел Янг, що працював у Національному бюро стандартів, придумав п’єзодвигун, що застосовується сьогодні в скануючих тунельних мікроскопах і для позиціонування наноінструментів з точністю до 0,01 ангстрем (1 нм = 10 А°).

1971 р. Рассел Янг висунув ідею приладу Topografiner, яка стала праобразом зондового мікроскопа. Настільки тривалі строки розробки подібних пристроїв пояснюються тим, що спостереження за атомарними структурами приводить до зміни їх стану, тому були потрібні якісно нові підходи, що не руйнують досліджувану речовину. Правда, незабаром роботи над Topografiner були припинені, і визнання до Янгу прийшло тільки в 1979 р., після чого він одержав безліч нагород.

1974 р. Японський фізик Норіо Танігучі, що працював у Токійському університеті, запропонував термін «нанотехнології» (процес поділу, складання й зміни матеріалів шляхом впливу на них одним атомом або однією молекулою), що швидко завоювала популярність у наукових колах.

1982 р. У Цюрихському дослідному центрі ІВМ фізики Герд Бінніг і Генріх Рорер (Нобелівські лауреати 1986 р. разом з Єрнстом Руской) створили скануючий тунельний мікроскоп (СТМ), що дозволяє будувати тривимірну картину розташування атомів на поверхнях провідних матеріалів.

1985 р. Троє американські хіміків: професор Райсського університету Річард Смелли, а також Роберт Карл і Херольд Крото (Нобелівські лауреати 1996 р.) відкрили фуллерени – молекули, що складаються із 60 атомів вуглецю, розташованих у формі сфери. Ці вчені також уперше зуміли виміряти об'єкт розміром 1 нм.

1986 р. Герд Бінніг розробив скануючий атомно-силовий зондовий мікроскоп, що дозволив нарешті візуалізувати атоми будь-яких матеріалів (не тільки провідних), а також маніпулювати ними.

1986 р. Американський учений Ерик Дрекслер, що працював у лабораторії штучного інтелекту Массачусетського технологічного інституту, написав книгу „Машини творення” („Engines of Creation”), у якій висунув концепцію універсальних молекулярних роботів, що працюють за заданою програмою і збирають що завгодно (у тому числі й собі подібних) з підручних молекул. Учений уже тоді досить точно передбачив чимало

прийдешніх досягнень нанотехнологій, і починаючи з 1989 р. його прогнози збуваються, причому нерідко зі значним випередженням строків.

1989 р. Учені Дональд Ейглер і Ерхард Швечер з Каліфорнійського наукового центру ІВМ зуміли викласти 35 атомами ксенону на кристалі нікелю назву своєї компанії. Для першого у світі цільового переносу окремих атомів у нове місце вони використовували СТМ виробництва ІВМ. Правда, такий напис проіснував недовго – атоми швидко „розбіглися” з поверхні.

1991 р. Японський професор Суміо Ліджима, що працював у компанії NEC, використовував фуллерени для створення вуглецевих трубок (або нанотрубок) діаметром 0,8 нм. На їхній основі в наш час випускаються матеріали в сто раз міцніші сталі. Залишалося навчитися робити такі трубки як можна більш довгими (зараз їх довжина складає одиниці міліметрів). Крім того, відкрилася можливість збирати з нанотрубок різноманітні наномеханізми із зачепленнями і шестернями.

1991 р. У США почала працювати перша нанотехнологічна програма Національного наукового фонду. Аналогічною діяльністю почав займатись й уряд Японії. А от у Європі серйозна підтримка таких досліджень на державному рівні почалася тільки з 1997 р.

1997 р. Ерік Дрекслер оголосив, що до 2020 р. стане можливою промислове складання нанопристроїв із окремих атомів. Дотепер майже всі його прогнози збувалися з випередженням.

1998 р. Сізій Деккер, голландський професор Технічного університету м. Делфтса, створив транзистор на основі нанотрубок, використовуючи їх як молекули. Для цього йому довелося першим у світі виміряти електричну провідність такої молекули. З'явилися технології створення нанотрубок довжиною 300 нм.

У Японії запущена програма „Astroboy” з розвитку наноелектроніки, здатної працювати в умовах космічного холоду й при жарі в тисячі градусів.

1999 р. Американські вчені – професор фізики Марко Рід (Єльський університет) і професор хімії Джеймс Тур (Райський університет) –

розробили єдині принципи маніпуляції як однією молекулою, так і їх ланцюжком.

Уряд США відкрив Національну нанотехнологічну ініціативу (NNI). У бюджеті США на цей напрямок виділено 270 млн. дол., комерційні компанії вклали в нього в 10 раз більше. 2001 р. Реальне фінансування NNI перевищило заплановане (422 млн. дол.) на 42 млн, а у 2002 р. 697 млн. дол., що на 97 млн більше запланованого [72].

4.4.2. Нанотехнології

Нанотехнології – це технології роботи з речовиною на рівні окремих атомів. Традиційні методи виробництва працюють із порціями речовини, що мають мільярди атомів. Це значить, що навіть найточніші прилади, зроблені людиною дотепер, на атомарному рівні виглядають як безладна мішанина. Перехід від маніпуляції з речовиною до маніпуляції окремими атомами – це якісний стрибок, що забезпечує безпрецедентну точність і ефективність.

Нанотехнології звичайно ділять на три напрямки:

- виготовлення електронних схем, елементи яких складаються з декількох атомів;
- створення наномашин, тобто механізмів і роботів розміром з молекулу;
- безпосередня маніпуляція атомами й молекулами й складання з них чого завгодно.

У практичному аспекті це технології виробництва пристроїв і їх компонентів, необхідних для створення, обробки й маніпуляції атомами, молекулами й частками, розміри яких перебувають у межах від 1 до 100 нанометрів. Однак, нанотехнологія зараз перебуває в початковій стадії розвитку, оскільки основні відкриття, що передвіщаються в цій області, поки не зроблені. Проте, проведені дослідження вже дають практичні результати. Використання в нанотехнології передових наукових результатів дозволяє віднести її до високих технологій.

4.4.3. Нанороботи

Нанороботи (також використовуються терміни „наноботи”, „ганоїди”, „наніти”) – роботи, створені з наноматеріалів і розміром, порівняним із розміром молекули (декількох нанометрів). Вони повинні мати функції руху, обробки й передачі інформації, виконання програм. Згідно із сучасними теоріями, нанороботи повинні вміти здійснювати двосторонню комунікацію: реагувати на акустичні сигнали й мати можливість підзаряджуватись або перепрограмуватися ззовні за допомогою звукових або електричних коливань. Також важливими представляються функції реплікації – самоскладання нових нанітів і програмованого самознищення, коли середовище роботи, наприклад, людське тіло, більш не має потреби в присутності в ньому нанороботів. В останньому випадку роботи повинні розпадатися на нешкідливі компоненти.

Чимало нанотехнологічних пристроїв уже створені й хоча вони поки є експериментальними розробками, практичні перспективи очевидні. Розроблений наноелектродвигун, що має обмотку довжиною з одну молекулу, здатну без втрат передавати струм. При подачі напруги починає обертатися ротор, що складається з декількох молекул. Існує також пристрій лінійного транспортування, здатний переміщати молекули на задану відстань. Розробляються також молекулярні біосенсори, антени, маніпулятори.

Сфера застосування нанороботів дуже широка. По суті, вони можуть бути необхідні при створенні, налагодженні й підтримці функціонування будь-якої складної системи. Наномашини можуть застосовуватися в електроніці для створення мініпристроїв або електричних кіл – дана технологія називається молекулярною нанообробкою. У перспективі будь-яке складання на заводі з компонентів може бути замінена простим складанням із атомів.

Однак на перше місце зараз вийшло питання застосування нанороботів у медицині. Тіло людини як би наштовхує на думку про нанороботів,

оскільки саме містить природні наномеханізми: лімфоцити і білі кров'яні тільця постійно функціонують в організмі, відновлюючи ушкоджені тканини, знищуючи шкідливі мікроорганізми й видаляючи сторонні частки з різних органів. Шляхом звичайної ін'єкції нанороботи можуть бути транспортовані в кров або лімфу. Для зовнішнього застосування розчин із цими роботами може бути нанесений на ділянку тканини. Одним з розроблених напрямків є транспортування ліків до уражених клітин. При звичайному введенні ліків лише одна молекула зі ста тисяч досягає мети, у той час як нанопристрій в білковій оболонці збільшує ефективність у сотні разів.

Нанороботи можуть робити буквально все: діагностувати стан будь-яких органів і процесів, втручатися в ці процеси, доставляти ліки, з'єднувати й руйнувати тканини, синтезувати нові. На даному етапі вченими розроблена складна програма, що моделює проектування й поведінку нанороботів в організмі. Надзвичайно детально розроблені аспекти маневрування в артеріальному середовищі, пошук білків за допомогою датчиків. Учені провели віртуальні дослідження нанороботів для лікування діабету, дослідження черевної порожнини, аневризми мозку, раку, біозахисту від отруйних речовин.

Існують, однак, і побоювання із приводу наномеханики. Так, згадана вище книга «Машини Творення» оповідає про збій у програмі роботів, у силу чого вони перетворюють усю землю в місиво із самих себе.

Серед найпоширеніших нанопристроїв на сьогоднішній день – нанотрубки. Вони відіграють різні ролі: від молекулярних фільтрів, що діють як звичайні сита, і до тривимірних шестірень, без яких важко уявити собі який-небудь механізм. Нанотрубки на малюнку майже цілком складаються з вуглецю, а точніше із замкнених графітових шарів. Зверніть увагу на виступи з боків трубок: саме вони виконують функції зубів, що перетворюють нанотрубки в шестірні (рис. 4.7) [73].

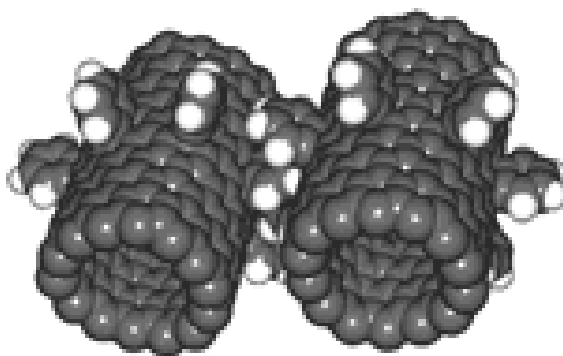


Рис. 4.7. Нонашестерні

4.5. Роботоситеми

4.5.1. Класифікація роботів за показниками, що визначають їхню конструкцію

До таких показників відносяться:

- тип приводів;
- вантажопідйомність;
- кількість маніпуляторів;
- тип і параметри їх робочої зони;
- мобільність і спосіб розміщення;
- виконання за призначенням.

Приводи, які використовуються в маніпуляторах і системах пересування роботів, можуть бути електричними, гідравлічними й пневматичними. Часто їх застосовують у комбінації. Наприклад, у ланках маніпулятора великої вантажопідйомності доречний гідравлічний привод, а його захватному пристрої - більш простий і малопотужний пневматичний привод.

Вантажопідйомність робота - це вантажопідйомність його маніпуляторів, а для транспортного робота ще і його шасі. Вантажопідйомність маніпулятора визначається масою переміщуваних ним об'єктів і залежно від призначення робота може становити від одиниць грам (надлегкі роботи, наприклад, застосовувані в мікроелектронній

промисловості) до декількох тисяч кілограм (надважкі, наприклад, транспортні й космічні роботи).

Кількість маніпуляторів у роботів у більшості випадків обмежено одним. Однак у залежності від призначення існують конструкції роботів з 2, 3 і зовсім рідко 4 маніпуляторами. Звичайно маніпулятори робота виконують однаковими, але є конструкції роботів і з різними маніпуляторами. Наприклад, існують промислові роботи для обслуговування пресів холодного штампування із двома різними маніпуляторами: один, основний, для захоплення заготовки й установки її в прес та інший, спрощеної конструкції, для виконання більш простої операції вилучення готової деталі.

Тип і параметри робочої зони маніпулятора визначають область навколишнього простору, у межах якої він може здійснювати маніпуляції, не пересуваючись, тобто при нерухомій основі. Робоча зона маніпулятора - це простір, у якому може перебувати його робочий орган при всіх можливих положеннях ланок маніпулятора. Форма робочої зони визначається системою координат, у якій здійснюється рух робочого органа маніпулятора, і числом ступенів вільності маніпулятора.

Мобільність робота визначається наявністю або відсутністю в нього системи пересування. У першому випадку роботи називають мобільними, а в другому - стаціонарними. Відповідно до призначення роботів у них застосовують системи пересування практично всіх відомих на сьогодні типів: від наземних (колісних, гусеничних тощо) до призначених для пересування у воді, повітрі й космосі. Специфічним способом пересування, що відноситься до робототехніки, є крокування.

По способу розміщення стаціонарні й мобільні роботи бувають такими, що знаходяться на підлозі, підвісними (мобільні роботи цього типу звичайно переміщаються по монорейках), роботи, що вбудовуються в інше обладнання (наприклад, з розміщенням на станині верстата, що обслуговується роботом).

Виконання робота за призначенням залежить від зовнішніх умов, у яких він повинен функціонувати. Розрізняють виконання нормальне, пилозахищене, теплозахищене, вологозахищене, вибухобезпечне тощо.

Класифікація робіт за швидкістю й точністю рухів. Ці параметри взаємозалежні й характеризують динамічні властивості роботів. Швидкодія маніпулятора визначається швидкістю переміщення його робочого органа й може бути розбита на 3 діапазони залежно від лінійної швидкості:

- мала - до 0,5 м/с;
- середня - від 0,5 до 1-3 м/с;
- висока - при більших швидкостях.

Для значної частини областей застосування роботів цей параметр дуже важливий, тому що визначає їхню продуктивність. Основна трудність при підвищенні швидкодії пов'язана з відомим протиріччям між швидкістю й точністю. Точність маніпулятора й системи переміщення робота характеризується похибкою позиціонування. Найчастіше точність робіт характеризується абсолютною похибкою. Точність робіт загального застосування можна розбити на 3 діапазони залежно від лінійної похибки:

- мала - від 1 мм і більше;
- середня - від 0,1 до 1 мм;
- висока - при меншій похибці.

Меншу точність мають роботи, призначені для виконання найбільш грубих, наприклад, транспортних рухів, а найбільшу, мікронну - роботи, використовуються, наприклад, в електронній промисловості. В порівнянні з людською рукою істотним недоліком сучасних роботів поки є зниження точності зі збільшенням ходу маніпулятора, у той час як у людини ці параметри в значній мірі розв'язані завдяки поділу рухів на грубі (швидкі) і точні.

Крім того, застосовуються різні, засновані на зазначених раніше параметрах, комбіновані й відносні показники. До них відносяться, зокрема, питома вантажопідйомність, віднесена до маси робота, вихідна потужність

маніпулятора (добуток вантажопідйомності на швидкість переміщення), віднесена до потужності його приводів; розмір робочої зони, віднесений до габаритів робота.

Іншими, раніше не згадуваними параметрами, що характеризують технічний рівень роботів, є число одночасно працюючих ступенів вільності, надійність, способи й час програмування [74].

4.5.2. Способи керування роботами

У робототехніці існують 3 завдання керування: керування маніпуляторами, системою пересування й спільне керування групою роботів. Застосовувані для розв'язку цих завдань способи керування можна класифікувати в такий спосіб.

За ступенем участі людини в процесі керування існують системи:

- автоматичного;
- автоматизованого;
- ручного керування.

За типом алгоритму автоматичного керування розрізняють системи:

- програмного;
- адаптивного;
- інтелектуального керування.

За типом руху існують системи керування:

- безперервні (контурні);
- дискретні позиційні (кроками „від точки до точки”);
- дискретні циклічні (з одним кроком по кожній координаті).

(У вітчизняній літературі дискретне позиційне керування часто називається просто позиційним.)

За видом керованих змінних розрізняють системи керування:

- положенням (позицією);
- швидкістю;
- силою (моментом);

Стосовно до керування маніпуляторами ці способи означають керування абсолютними координатами його робочого органа його швидкістю й силою. Часто перераховані способи керування застосовуються в комбінації одночасно по різних координатах, у вигляді деякої функціональної залежності однієї керованої змінної від іншої.

Керування рухом по окремих ступенях вільності може бути **безперервним і дискретним**. В останньому випадку керування рухом здійснюється завданням послідовності точок із зупинкою в кожній з них. Найпростішим варіантом дискретного керування є циклічне, при якому кількість точок позиціонування по кожній ступені вільності мінімальна - найчастіше обмежена двома – початковою та кінцевою [75].

4.5.3. Основні функції робота

На початку 1960-х років почалися дослідження, пов'язані з керуванням механічною рукою за допомогою комп'ютера, а також дослідження штучного зору, заснованого на спільному використанні телекамери й комп'ютера. Приблизно в 1970 р. був розроблений прототип інтелектуального робота, складеного з елементів штучного зору й руки. Результати цих досліджень поширилися на роботи, призначені для використання в промисловості й названі промисловими роботами.

Слід зазначити, що роботи, створювані інженерами, відрізняються від роботів, що описуються у романах і створюються умільцями. У своїй більшості вони не схожі на людину. Роботи в основному приймають вид корисних машин з урахуванням тих функцій, які може виконувати людина. До функцій, які в першу чергу повинен мати робот, можна віднести наступні чотири.

Перша функція - **функція маніпулювання й переміщення**. Маніпулюючи предметами, робот повинен виконувати корисну роботу й забезпечувати своє переміщення. Другою функцією є **функція визначення стану зовнішнього середовища**. Робот повинен мати органи чуття, що

дозволяють йому «бачити», «сприймати дотиком», «відчувати» силу, а також «розпізнавати» об'єкти. До третьої функції відноситься **функція осмислення й прийняття рішення**. Вона дає можливість планувати послідовність операцій, необхідних для досягнення мети роботи, заданої людиною, а також для здійснення відповідного керування. І нарешті, четвертою функцією є **функція діалогу**, при реалізації якої забезпечується ефективний зв'язок людини з роботом. За допомогою комплексної реалізації цих чотирьох функцій можна створити робот.

У сучасній літературі, присвяченій роботам, докладно розглядаються проблеми й основні складові елементи робота. У їхньому числі - пристрої роботів, керування маніпуляторами, обробка візуальної інформації, мови роботів, штучний інтелект, застосування роботів у промисловості тощо. Звичайно, наявність окремих технічних елементів є необхідною умовою для створення робота, однак далеко не достатньою для створення ідеального робота як інтелектуальної машини, здатної виконувати найрізноманітніші роботи. Необхідний також такий підхід, при якому різні функції, формалізовані у своїй єдності, синтезуються в одній системі.

При розробці робота як системи необхідно враховувати два аспекти, що взаємно доповнюють один одного. По-перше, цільову спрямованість. Роботи проектують як системи відповідно до їхнього застосування й поставленими перед ними цілями. При цьому необхідно забезпечити роботам відповідну більш високу універсальність. А якщо ні, то вийде автомат, яким користувалися раніше. При цільовій спрямованості з урахуванням потреб на універсальність робота доводиться накладати деякі обмеження і як у випадку системи здійснювати деякий баланс. Як приклад розглянемо робот, що використовується при складанні. Можна вважати, що доцільно мати такий робот, який буде здійснювати складальні роботи із застосуванням деякого набору типізованих виробів. Виходячи із цього, можна внести відповідні обмеження на його універсальність. При такому підході слід чітко представляти мету проектування складального робота, упорядкувати функції

його елементів і, виключивши зайве, здійснити оптимізацію його структури, для того щоб вона відповідала поставленим цілям. У більшості випадків цим аспектом керуються при створенні роботів, призначених для використання в промисловості.

По-друге, аспект, орієнтований на універсалізацію. При цьому прагнуть розробити таку логіку системи, при якій пріоритетними є спільність і розширюваність. Ці два поняття взаємно доповнюють один одного. При реалізації цільової спрямованості слід починати з конкретизації мети. З погляду практики це цілком природно, і система виходить більш доботною. Однак необхідно мати на увазі, що якщо оптимізація здійснена для якогось конкретного випадку, то звичайно буває важко використовувати систему в іншому випадку. На підставі численних прикладів використання цілеспрямованого підходу важливо індуктивно виявляти такі основні функції роботів, які мають загальний характер. Припустимо, що на базі безлічі основних функцій, установлених індуктивним шляхом, удалося побудувати робот, який можна розглядати як логічну систему, що володіє великою загальністю і розширюваністю. Тоді, як видно, в окремому випадку з ним можна поводитись як з виробом, що одержують із універсальної системи. Таким чином, можна ефективно розробляти роботи, що відповідають найрізноманітнішим вимогам.

Слід звернути увагу на те, що серед машин, дотепер створених людиною, найбільш універсальними системами є обчислювальні машини. Це - системи, для яких навіть розроблювачі не можуть установити границі виконуваних функцій. Розробивши відповідні програми, за допомогою порівняно невеликого числа ретельно відібраних команд (шляхом складання їх різних комбінацій) на ЕОМ можна проводити не тільки числові розрахунки, але й обробляти буквену, символічну й графічну інформацію, управляти машинами тощо. Теоретично можливості апаратного забезпечення ЕОМ можна виразити у вигляді набору команд. Роль програмного забезпечення зводиться до того, щоб певним чином поєднувати команди,

створювати програми і в остаточному підсумку здійснювати конкретну реалізацію систем різного призначення. Можна дійти висновку про те, що універсальність ЕОМ зводиться до розробки системи ретельно підібраних команд і системи програм, що представляють собою комбінацію цих команд.

До ідеальної роботи можна віднести таку універсальну машину, яка має руки, здатні виконувати різноманітну роботу, має очі, що дозволяють пізнавати предмети навколишнього середовища, має певний інтелект за допомогою якого робот може сам визначати й керувати послідовністю виконання операцій, а також забезпечити природне спілкування (діалог) з людиною, наказам якого він (робот) повинен підкорятися.

Як видно, універсальність роботи можна охарактеризувати безліччю основних функцій, що відносяться до розпізнавання та дії, а також системою відповідних до цих функцій програм, що володіють великою розширюваністю. Імовірно, стосовно команд ЕОМ команди роботів повинні бути макрокомандами. Їхніми одиницями можуть служити основні сенсорні функції й основні дії [7].

4.5.4. Манипуляційні системи роботів

Сьогодні основним типом манипуляційних систем роботів є механічні маніпулятори. Вони являють собою просторові механізми у вигляді кінематичних ланцюгів з ланок, що утворюють кінематичні пари з кутовим або поступальним відносним рухом і системою приводів звичайно роздільних для кожного ступеня вільності. Маніпулятори закінчуються робочим органом.

Ступені вільності маніпулятора діляться на переносні й орієнтуючі. Переносні ступені вільності служать для переміщення робочого органа в робочій зоні маніпулятора, а орієнтуючі - для його кутової орієнтації. Мінімумально необхідне число переносних ступенів вільності для переміщення робочого органа в просторі робочої зони рівно 3. Однак для розширення манипуляційних можливостей і реалізації більш складних траєкторій руху,

наприклад, з обходом перешкод, а також для підвищення швидкодії маніпулятори звичайно проектують з декількома надлишковими переносними ступенями вільності, хоча це, ускладнює й здорожчує робота. Сучасні маніпулятори в середньому мають 4-6 ступенів вільності, але існують маніпулятори й з 8-9 такими ступенями.

Максимально необхідне число орієнтуючих ступенів вільності рівно 3. Вони реалізуються кінематичними парами з кутовим переміщенням, що забезпечують поворот робочого органа маніпулятора щодо його поздовжньої й двох інших взаємно перпендикулярних осей.

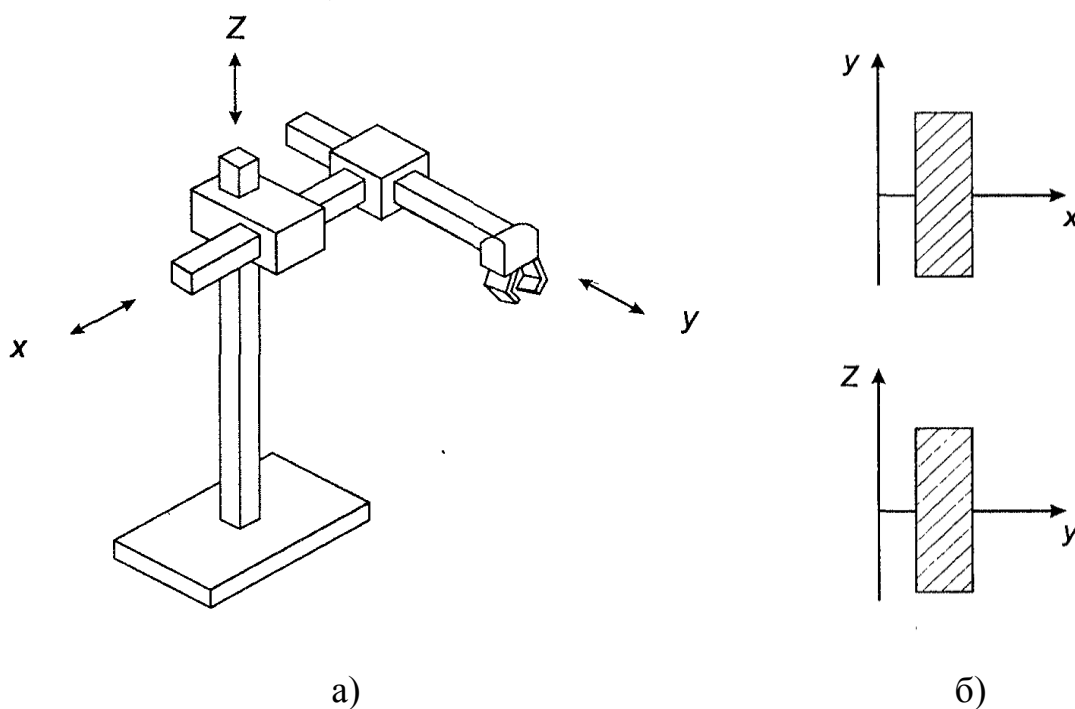


Рис. 4.8. Маніпулятор із прямокутною системою координат (а) і його робоча зона (б)

Маніпулятори, що працюють у прямокутній системі координат (рис. 4.8), мають робочу зону у формі паралелепіпеда. Тут усі переміщення тільки поступальні. Тому така система координат найбільш зручна для виконання прямолінійних рухів. Крім того, вона максимально спрощує програмування робота, тому що воно звичайно виконується саме в прямокутній системі координат, і, отже, у цьому випадку не потрібно перерахування програм з однієї системи координат в іншу.

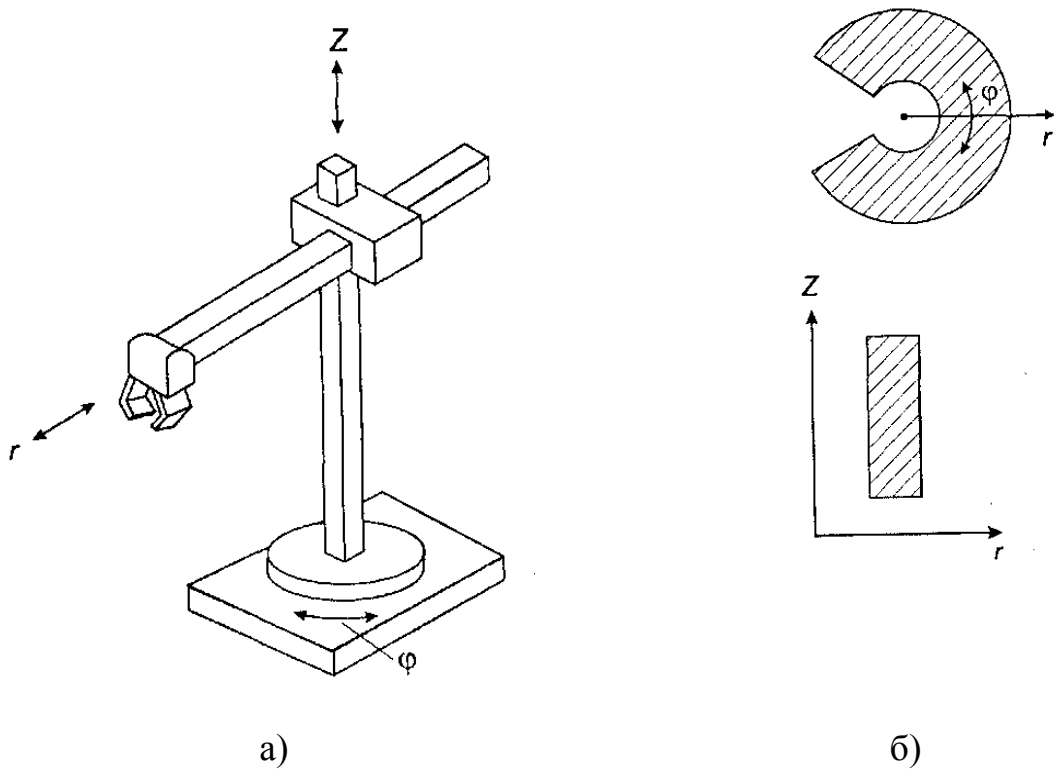


Рис. 4.9. Маніпулятор із циліндричною системою координат (а)
та його робоча зона (б)

У маніпуляторах із циліндричною системою координат (рис. 4.9) поряд з поступальними переміщеннями здійснюється одне кутове переміщення (по колу). Відповідно, робоча зона обмежена циліндричними поверхнями.

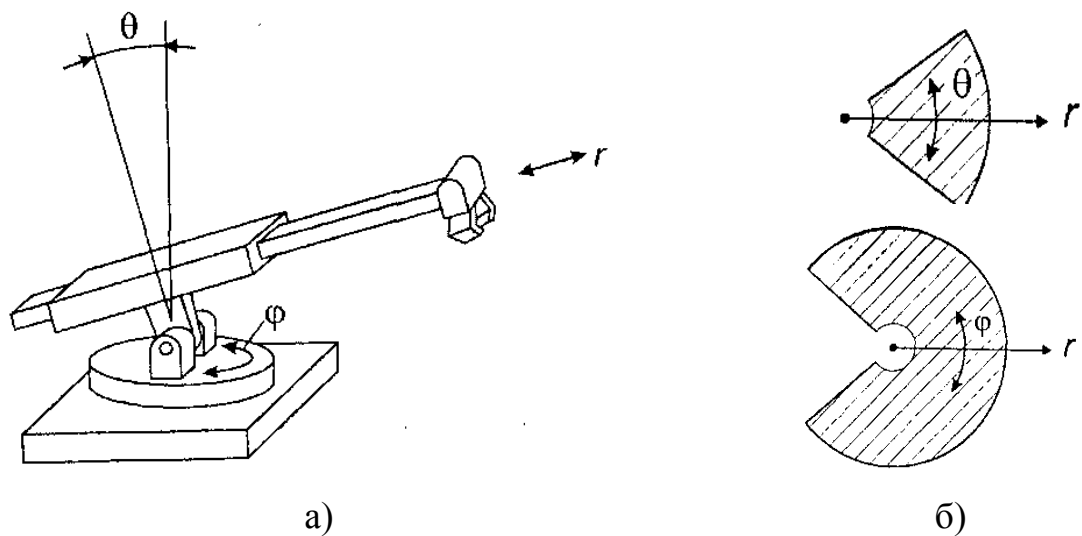


Рис. 4.10. Маніпулятор із сферичною системою координат (а)
та його робоча зона (б)

У сферичній системі координат (рис. 4.10) здійснюються вже 2 кутових переміщення й робоча зона обмежена сферичними поверхнями. Маніпулятори з такою системою координат, як правило, складніші, ніж із циліндричною системою, однак компактніші.

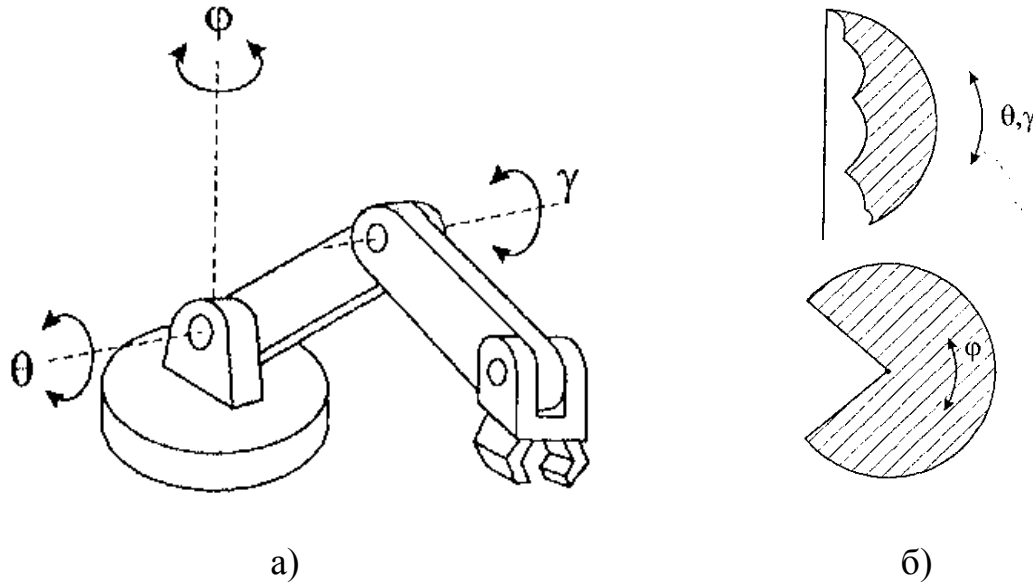


Рис. 4.11. Маніпулятор з кутовою системою координат (а)
та його робоча зона (б)

Показаний на рис. 4.11 маніпулятор з кутовою системою координат робить тільки кутові переміщення, тобто всі його ланки являють собою шарніри. Тому часто такі маніпулятори називають шарнірними й антропоморфними. Роботи з такого типу маніпуляторами завдяки можливості останніх складатися, не виступаючи практично за габарити підстави робота, мають найбільшу компактність, хоча й найбільш складні в керуванні.

Представлені на рис. 4.1-4.4 маніпулятори мають усього по 3 переносних ступенів вільності рухливості. Однак оскільки маніпулятори реальних роботів містять більше число ступенів вільності, у них часто використовуються різні комбінації розглянутих раніше основних типів систем координат [74].

4.5.5. Сенсорні системи робота (підсистеми тактильного та зорового чуття)

Підсистема тактильного чуття

У робота (рис. 4.12) два пальці при стисненні й розтискуванні залишаються паралельними один одному. На внутрішніх поверхнях розміщені чуттєві елементи (датчики), які реагують на дотик. У зап'ястя робота вмонтований 6-вісний датчик відчуття за зусиллями, які діють у напрямках x , y , z і моментів m_x , m_y , m_z , діючих відносно відповідних осей (рис. 4.12).

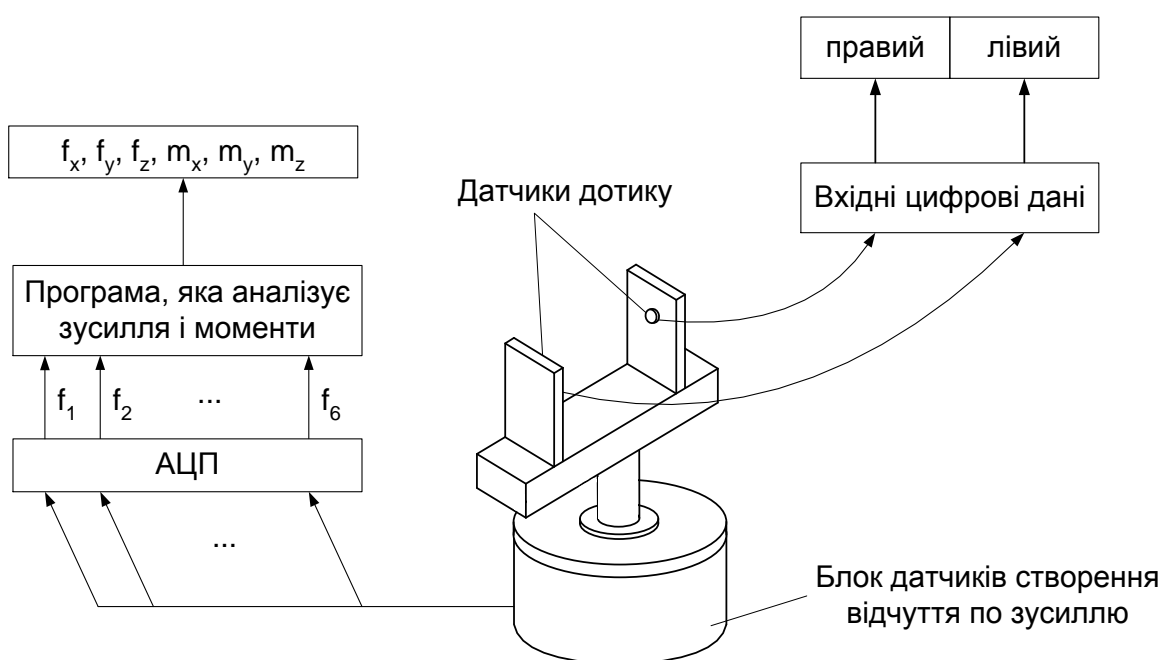


Рис. 4.12. Тактильні відчуття та відчуття за зусиллям робота

Підсистема технічного зору

У якості вхідного пристрою, який забезпечує роботу „зору”, зазвичай використовують телевізійну камеру, яка складається з передавальної трубки й твердотілого формувача сигналу зображення. Використовуючи квантування сигналу відображення, котрий отримують при скануванні поверхні відображення (зображення) формують цифрові сигнали, які можна обробляти на комп'ютері. За допомогою такої обробки можна здійснювати розпізнавання зорових образів.

Існують способи, котрі дозволяють шляхом сполучення різних параметрів виражати ознаки плоских фігур. У порівняно простих системах технічного зору, які зараз випускаються промисловістю, дуже часто для розпізнавання об'єктів проводять розрахунки значень дескрипторів ознак фігур і використовуються параметри, що представляють собою сполучення цих значень. Згідно системи технічного зору SRI, існують такі типові дескриптори (discription – опис, to discribe – описувати): N – кількість отворів у фігурі; S – площа фігури; L – периметр фігури; X_c, Y_c – координати центра фігури; R_{max} – max радіус (максимальна відстань від центру фігури до точки, розміщеної на периметрі); R_{min} – min радіус (мінімальна відстань від центру фігури до точки, розміщеної на периметрі фігури); R_{cp} – середній радіус (середня відстань від центру фігури до точки, розміщеної на периметрі); α_{max} – кут між max радіусом та віссю X ; α_{min} – кут між min радіусом та віссю X ; length – довжина більшої сторони описаного прямокутника, сторони котрого паралельні головним вісям інерції; width – довжина меншої сторони описаного прямокутника, сторони котрого паралельні головним вісям інерції; $X_{max}, X_{min}, Y_{max}, Y_{min}$ – максимальні/мінімальні значення проєкцій фігури на напрямки вісей X та Y відповідно.

Якщо об'єкти, які розпізнаються, відомі заздалегідь, то, сполучаючи різноманітні основні дескриптори ознак, можна визначити параметри класифікації розглядуваних об'єктів. Такі параметри носять назву факторів форми.

Розглянемо класифікацію кола, шестикутника й квадрата (рис. 4.13). У даному випадку у якості фактору форми можна обрати, наприклад, відношення площі фігури (S) до площі описаного прямокутника ($S/\text{довжина} \times \text{ширина}$) і відношення ширини описаного прямокутника до його довжини (табл. 4.2). Керуючись значеннями цих факторів форми, можна легко розпізнати вказані фігури.

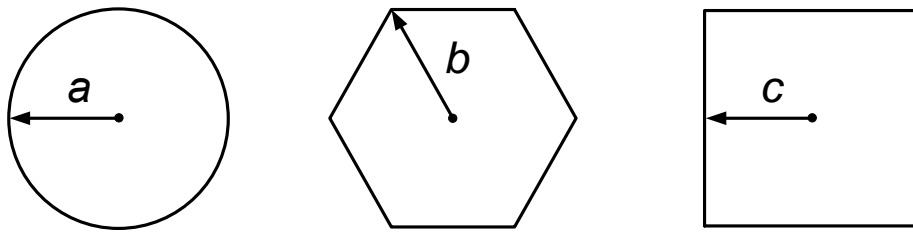


Рис. 4.13. Основні параметри простих геометричних фігур

Таблиця 4.2. Розпізнавання фігур за фактором форми

Параметри		Найменування фігури		
		круг	шестикутник	квадрат
Основні ознаки	Довжина кола (параметр L)	$2\pi a$	$6b$	$8c$
	Площа (S)	πa^2	$\frac{3\sqrt{3}}{2}b^2$	$4c^2$
	Ширина	$2a$	$\sqrt{3}b$	$2c$
	Довжина	$2a$	$2b$	$2c$
Фактор форми	S/довжина*ширина	$\frac{\pi}{4} = 0,785$	$3/4=0,75$	1
	Ширина/довжина	1	$\frac{\sqrt{3}}{2} = 0,865$	1

Буває необхідно класифікувати для розпізнавання й численні більш складні фігури, однак при використанні факторів форми та їх значень необхідна особлива винахідливість у складанні процедури прийняття рішення при розпізнаванні образів.

Для фігур, котрі необхідно розпізнавати, можна заздалегідь створити зображення. Такі зображення носять назву шаблонів. Щоб серед поступаючих зображень проаналізувати існування чи відсутність зображень, що співпадають з шаблонами, можна скористатись примітивним способом,

який полягає у безпосередньому зіставленні вхідних зображень з шаблонами. Цей спосіб називається підбором шаблонів.

При такому підході треба: 1) нормалізувати зображення за величиною; 2) виконати відповідні операції, пов'язані із забезпеченням співпадіння центрів ваги та напрямків осей симетрії.

При зіставленні шаблона вхідне зображення можна переміщувати паралельно самому собі, повертати, змінювати масштаб тощо. Однак це вимагає часу на пошук а ефективність зіставлення низька!

Дискриптори основних ознак включають у себе положення центру ваги фігури, напрямки головних осей інерції, описаний квадрат та ін. Якщо при паралельних переміщеннях та поворотах використовувати подібні дискриптори, то ефективність зіставлення зображень та шаблонів набагато вища!

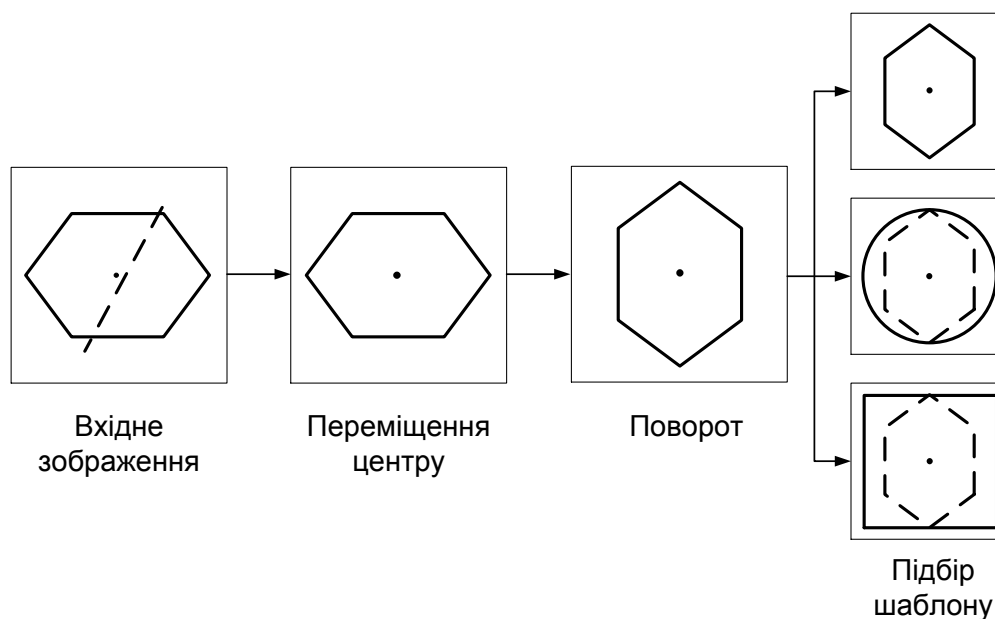


Рис. 4.14. Приклад узгодження з еталоном

Алгоритм узгодження:

- 1) спочатку здійснюють паралельний перенос так, щоб центр ваги вхідного зображення співпав з центром площини креслення;
- 2) повертають зображення так, щоб головна вісь інерції розміщувалась перпендикулярно;

3) порівнюють вхідне зображення з 3-ма шаблонами.

При співставленні використовується операція, котра дозволяє на основі отриманого числа співпадінь елементів зображення зробити висновок про те, який шаблон підходить.

Вимірювання відстаней. За допомогою однієї лише телекамери неможливо виміряти положення предмету у тривимірному просторі. Такі вимірювання можна виконати, наприклад, за допомогою сумісного використання телекамери й лазерного скануючого пристрою з біжучим променем. У цьому пристрої є 2 маленьких дзеркальця, змінюючи нахил котрих можна посилати промінь лазера у будь-якому напрямку. За допомогою телекамери можна виміряти центральні координати невеликої плями, яка утворюється при падінні лазерного променя на тіло. Якщо відомі кут відхилення лазерного променя і координати зображення плями на сприймаючій поверхні телекамери, то можна розрахувати положення плями у тривимірному просторі на поверхні тіла, яке опромінюється [7].

4.5.6. Циклічне програмне керування

Найпоширеніший тип роботів із циклічним програмним керуванням - це пневматичні промислові роботи. Пневматичні приводи роботів мають циклічне керування, тобто однокрокове релейне керування без зворотних зв'язків. Зупинка привода здійснюється за допомогою механічного упору з кінцевим вимикачем і демпфера на рухомій частині привода.

Пристрій керування робота реалізує загальносистемний рівень керування, завдання якого здійснити координацію роботи приводів. Остання полягає в установці визначеної послідовності включення окремих приводів і необхідності витримок часу між цими моментами.

На рис. 4.15 показаний процес циклічного програмного керування приводом. Він складається з 3 етапів: розгону, руху з постійною швидкістю й гальмування. Тривалість другого етапу - усталеного руху з постійною швидкістю визначається величиною заданого переміщення привода. Якщо ця

Особливо важливо враховувати пружні деформації, які створюють коливання у циклічних приводах при їхньому розгоні й гальмуванні, тобто тоді, коли з'являються прискорення. Найбільш небажані коливання, що виникають при гальмуванні привода, тому що вони затягують процес його зупинки і знижують швидкодію робота. Коливання, що виникають на етапі розгону привода, звичайно встигають загаснути до моменту його гальмування й тому менш шкідливі. Однак, якщо тривалість етапу руху з постійною швидкістю відсутня обидва ці типи коливань можуть накладатися, що ще більш утруднить процес гальмування.

Оскільки при циклічному керуванні гальмування привода здійснюється демпфером при наїзді на упор, то гасіння можливих коливань - завдання настроювання демпфера, а саме добору величини створюваної їм сили в'язкого тертя.

Циклічне керування - це найпростіший спосіб однокрокового дискретного керування, однак на цій підставі його не слід розглядати як вироджений варіант багатокрокового дискретного позиційного керування. Системи циклічного керування принципово відрізняються від позиційних способів позиціонування за допомогою механічних упорів. Це забезпечує циклічним системам значно більшу точність і швидкодію.

Що стосується швидкодії, то, як впливає із теорії оптимальних систем, саме релейне керування забезпечує для об'єктів типу виконавчих систем роботів максимальну швидкодію.

4.5.7. Системи адаптивного керування

Розглянуті раніше системи програмного керування реалізують найпростіший спосіб керування, коли апріорно досить точно відомо все про зовнішнє середовище й про самого робота, що дозволяє заздалегідь спланувати і запрограмувати роботу робота. Іншими словами, таке керування припускає повністю детерміновані зовнішні й внутрішні умови. Декілька типових прикладів, коли це не виконується:

- операція захоплення довільно розташованого об'єкта, що переміщується (для її виконання необхідно автоматичне наведення захвату маніпулятора на об'єкт);
- нагвинчування гайки на болт (для цього потрібно попередньо „наживити” гайку на різьблення);
- дугове зварювання швів, положення яких може варіюватися (потрібно вести зварювальну головку строго по шву й з певною кутовою орієнтацією);
- вставляння стрижня в отвір з точністю, яка перевищує точність позиціонування при програмному керуванні, і тому вимагає пробних пошукових рухів;
- рух у заздалегідь невідомому просторі з обходом непередбачених перешкод.

Для виконання подібних операцій необхідно оперативно одержувати інформацію про середовище безпосередньо в ході їх виконання для використання її в реальному часі при керуванні рухом. Таке керування у функції від поточної інформації про зовнішнє середовище називається адаптивним. Необхідною умовою його реалізації є наявність сенсорних систем, що дають цю інформацію. Адаптація розширює функціональні можливості робота при виконанні завдань, що не розв'язуються програмно, і спрощує його експлуатацію, не вимагаючи точного знання моделі зовнішнього середовища й самого робота.

Адаптивне керування можливе в наступних варіантах:

- визначення значень заздалегідь невідомих параметрів для введення їх у керуючу програму системи. Тут адаптивне керування є надбудовою над системою програмного керування, здійснюючи її самонастроювання. Це параметрична адаптація.
- вибір керуючої програми з готового набору програм або складання її з набору типових підпрограм, у тому числі в ході виконання операції, на

основі оцінки поточної ситуації за сенсорною інформацією. Тут адаптивне керування теж доповнює систему програмного керування.

- зміна (перемикання) структури системи керування, тобто її алгоритму, на основі оцінки поточної обстановки. Наприклад, зміна способу корекції в системі програмного керування або перехід від однієї сенсорної системи до іншої в системі адаптивного керування. Це структурна адаптація (самоорганізація системи).
- безпрограмне керування рухом у реальному часі тільки за сенсорною інформацією. Наприклад, виконання операції наведення робочого органа маніпулятора на довільно розташований об'єкт.

Таким чином, адаптивне керування може здійснюватися як через систему програмного керування приводами, так і шляхом безпосереднього впливу на останні.

При виконанні конкретних технологічних операцій адаптивне керування найчастіше застосовується в комбінації із програмним, забезпечуючи зміну структури системи програмного керування й підстроювання її параметрів або складання керуючої програми, або, нарешті, зміну її, коли необхідно повністю переходити на керування по поточній сенсорній інформації.

Крім того, можлива комбінація перерахованих способів адаптації, у тому числі й ієрархічна, коли є кілька рівнів адаптації („адаптація адаптації”). Наприклад, над рівнем параметричної адаптації, що здійснює підстроювання системи програмного керування, може перебувати рівень структурної адаптації, що здійснює дискретну зміну в схемі останньої в міру вичерпання можливості підстроювання

У системах спільного керування приводами маніпулятора застосовуються ті ж способи адаптації, що й при керуванні окремими приводами. На рис. 4.16 показана схема системи адаптивного керування маніпулятором.

Відповідно до цієї схеми адаптивне керування може відбуватись за трьома варіантами:

- в функції від оцінюваної якості керування з метою стабілізації, тобто усунення відхилення показника якості (наприклад, точність виведення робочого органу на задану позицію) від заданого значення або шляхом пошуку й підтримки оптимального значення обраного показника якості керування;
- з тією ж метою по відхиленню від еталонної моделі;
- в функції від вимірюваних сенсорними системами параметрів об'єктів зовнішнього середовища або координат робочого органу маніпулятора щодо об'єктів, з якими він взаємодіє, або від зусилля взаємодії з ними.



Рис. 4.16. Схема адаптивного керування маніпулятором

Зупинимося на останньому варіанті адаптивного керування рухом оскільки воно характерно саме для зовнішньої адаптації. Особливість алгоритмів такого керування - наявність у них логічних операцій поряд з керуванням рухом. На рис. 4.14 наведено приклад алгоритму такого керування для виконання типової операції - пошук і захоплення довільно розташованого предмету.

Для виконання таких операцій захват оснащується сенсорами у вигляді далекомірів, розташованих на пальцях, і датчиком наявності предмета між пальцями. На рис. 4.17 наведена структурна схема алгоритму захвату маніпулятором довільно розташованого на площині в робочій зоні предмета. Після виводу захвату по програмі в зону очікуваного знаходження предмета проводиться сканування захватом робочої площини у околі точки виводу (наприклад, рядкове сканування у квадраті із центром у цій точці). Захват при цьому орієнтується торцями пальців перпендикулярно до робочої площини. При виявленні сенсорами предмета захват зупиняється над ним і орієнтується так щоб його пальці перебували симетрично щодо середини предмета із двох сторін від його найбільш вузької частини. Після цього захват опускається й по сигналу датчика наявності предмета пальці захвата зводяться, захоплюючи предмет. У випадку невдачі маніпулятор зупиняється й сигнал про це надходить на пульт оператора. Однак звичайно до цього проводиться повторення операції.

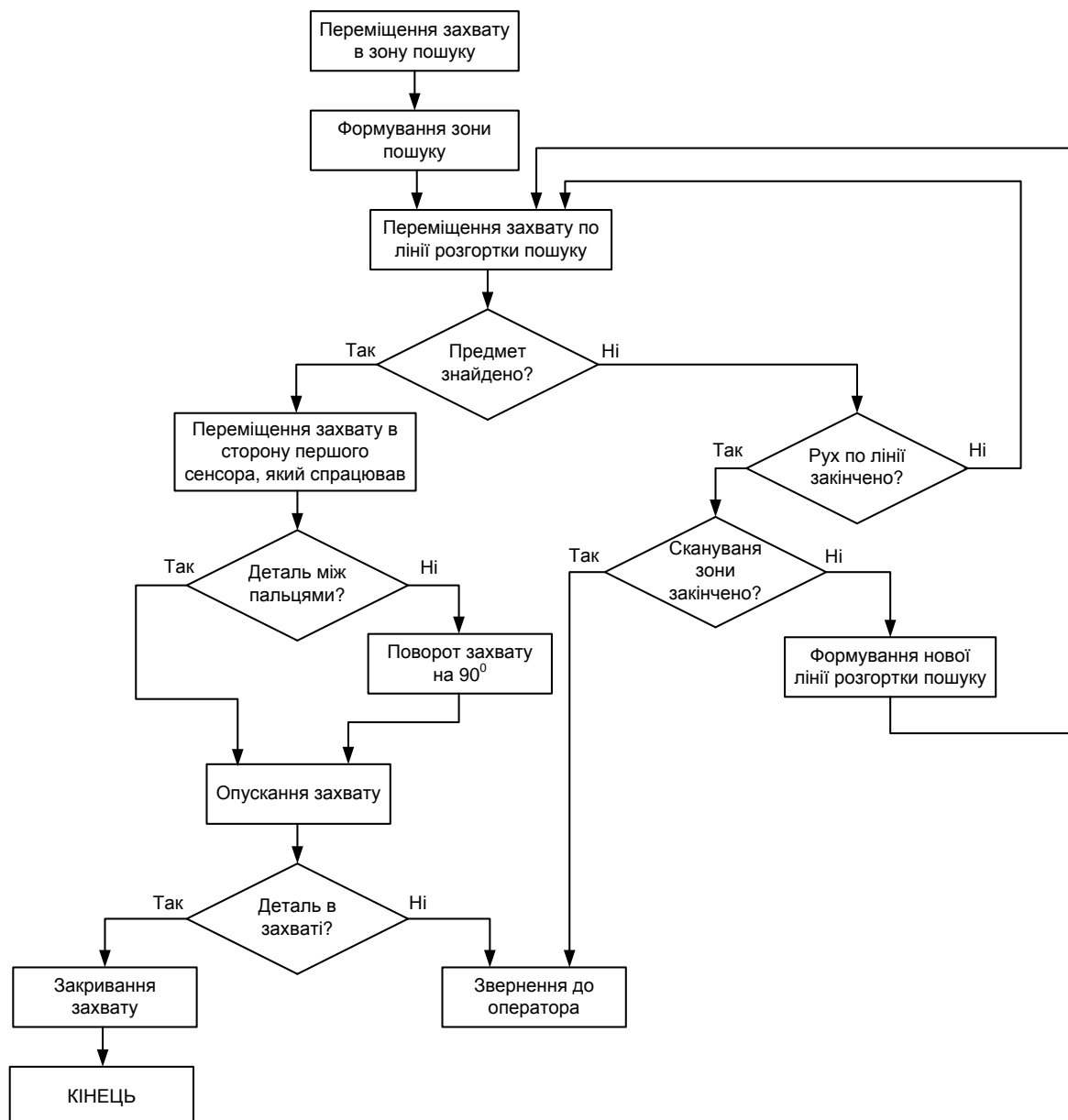


Рис. 4.17. Структурна схема пошуку і захоплення предмету

4.5.8. Дві задачі кінематики маніпуляторів

У робототехніці, є два основні задачі кінематики: пряма й зворотна. Розглянемо ці задачі на стандартному прикладі маніпулятора.

Пряма задача - це обчислення положення (X, Y, Z) робочого органа маніпулятора (кінцевої точки маніпулятора) по відомій його кінематичній схемі й заданій орієнтації $(\alpha_1, \alpha_2, \dots, \alpha_n)$ його ланок (n - число ступенів вільності маніпулятора, α - кути повороту).

Зворотна задача - це обчислення кутів ($\alpha_1, \alpha_2 \dots \alpha_n$) по заданим значенням координат (X, Y, Z) положення робочого органа й відомій схемі його кінематики.

Розв'язок прямої задачі говорить - де буде перебувати робочий орган маніпулятора, при заданих кутах його суглобів, а зворотна задача, навпаки, говорить: як потрібно „вивернутися” маніпулятору, щоб його робочий орган виявився в заданому положенні.

Очевидно, що більш розповсюдженою та важливою є саме зворотна задача кінематики. Але потрібно мати на увазі, що ця задача рідко може бути розв'язана однозначно. Справа в тому, що хоча для кутів ($\alpha_1, \alpha_2 \dots \alpha_n$) завжди існує єдине положення (X, Y, Z) робочого органа, але не факт, що для положення (X, Y, Z) віднайдеться така ж єдина комбінація кутів ($\alpha_1, \alpha_2 \dots \alpha_n$). Швидше за все, досягти заданого положення (X, Y, Z) можливо й при іншій комбінації кутів ($\acute{\alpha}_1, \acute{\alpha}_2 \dots \acute{\alpha}_n$). При розв'язуванні зворотної задачі аналітично, ця неоднозначність проявляється в явному виді (наприклад, через наявність квадратних коренів).

Розглянемо приклад: у нас є маніпулятор, здатний працювати тільки в одній площині, що й має два шарніри і дві ланки (рис. 4.18).

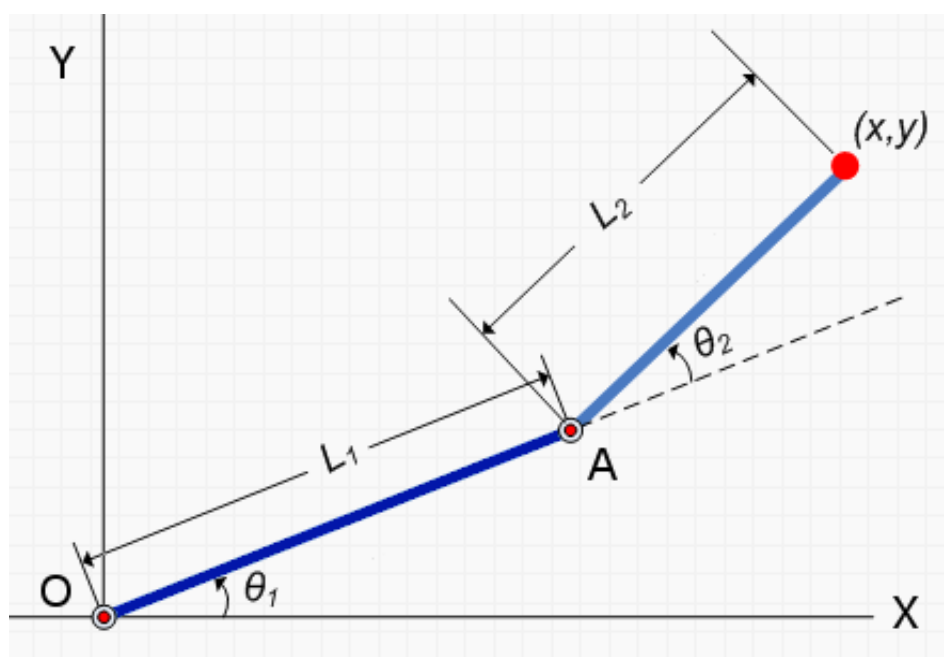


Рис. 4.18. Кінематична схема маніпулятора

Перша ланка L_1 закріплена на основі й повернута на кут θ_1 , друга ланка L_2 кріпиться до кінця першої ланки й повернута до неї на кут θ_2 . Робочий орган маніпулятора перебуває на кінці другої ланки .

Пряма задача кінематики полягає в знаходженні координат робочого органа (x, y) по заданим $L_1, L_2, \theta_1, \theta_2$.

Розв'язок: тут, ми маємо дві системи відліку - перша, пов'язана із точкою кріплення ланки L_1 - O (глобальна система координат), а друга - з початком координат у точці A кріплення другої ланки (локальна система координат).

Знайдемо зсув другої системи відносно першої (координати точки A у системі відліку O):

$$\begin{cases} \tilde{\delta}_A = L_1 \cos \theta_1; \\ y_A = L_1 \sin \theta_1. \end{cases} \quad (4.4)$$

Координати робочого органу (x, y) у локальній системі координат:

$$\begin{cases} \delta' = L_2 \cos(\theta_1 + \theta_2); \\ y' = L_2 \sin(\theta_1 + \theta_2). \end{cases} \quad (4.5)$$

Додаючи відповідні відстані маємо:

$$\begin{cases} x = x_A + \delta' = L_1 \cos \theta_1 + L_2 \cos(\theta_1 + \theta_2); \\ y = y_A + y' = L_1 \sin \theta_1 + L_2 \sin(\theta_1 + \theta_2). \end{cases} \quad (4.6)$$

Тепер, розглянемо приклад зворотної задачі кінематики. Той же малюнок, але тепер потрібно знайти такі кути θ_1 і θ_2 , які дозволять маніпулятору із довжинами ланок L_1 і L_2 перемістити робочий орган у задану точку (x, y) .

Проведемо пряму B (рис. 4.19), що з'єднує початок координат O із заданою точкою (x, y) .

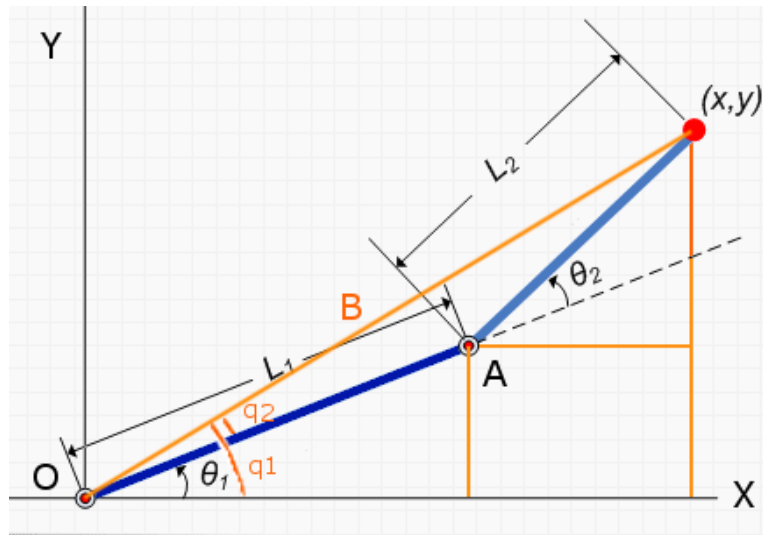


Рис. 4.19. До розв'язку зворотної задачі кінематики

Маємо:

$$\begin{cases} B^2 = x^2 + y^2; \\ x = B \cos q_1; \\ y = B \sin q_1. \end{cases} \quad (4.7)$$

де q_1 — кут між віссю OX і прямою B; q_2 — кут між прямою B і ланкою L_1 .

Звідси:

$$\theta_1 = q_1 - q_2, \quad (4.8)$$

$$q_1 = \arccos \frac{x}{B}. \quad (4.9)$$

Або

$$q_1 = \operatorname{arctg} \frac{y}{x}. \quad (4.10)$$

q_2 знаходимо за допомогою теореми косинусів (для плоского трикутника зі сторонами a, b, c і кутом α , що лежить навпроти сторони a , справедливе співвідношення:

$$a^2 = b^2 + c^2 - 2bc \cos \alpha. \quad (4.11)$$

У нашому випадку, по теоремі косинусів:

$$L_2^2 = B^2 + L_1^2 - 2BL_1 \cos q_2. \quad (4.12)$$

Звідси:

$$q_2 = \arccos\left(\frac{L_1}{2B} + \frac{B}{2L_1} - \frac{L_2^2}{2BL_1}\right), \quad (4.13)$$

$$\theta_1 = q_1 - q_2 = \arccos\left(\frac{x}{B}\right) - \arccos\left(\frac{L_1}{2B} + \frac{B}{2L_1} - \frac{L_2^2}{2BL_1}\right).$$

По тій же теоремі косинусів знайдемо кут θ_2 : як видно з рис. 4.19 кут θ_2 можна знайти, якщо від 180° відняти кут $\angle OAx$. Отже, $\angle OAx$ визначається так:

$$\angle OAx = \pi - \theta_2, \quad (4.14)$$

$$\hat{A}^2 = L_1^2 + L_2^2 - 2L_1L_2 \cos(\pi - \theta_2), \quad (4.15)$$

$$\theta_2 = \pi - \arccos\left(\frac{L_1}{2L_2} + \frac{L_2}{2L_1} - \frac{2B^2}{L_1L_2}\right). \quad (4.16)$$

Очевидно, що ланки можна розташувати й по-іншому (рис. 4.20)

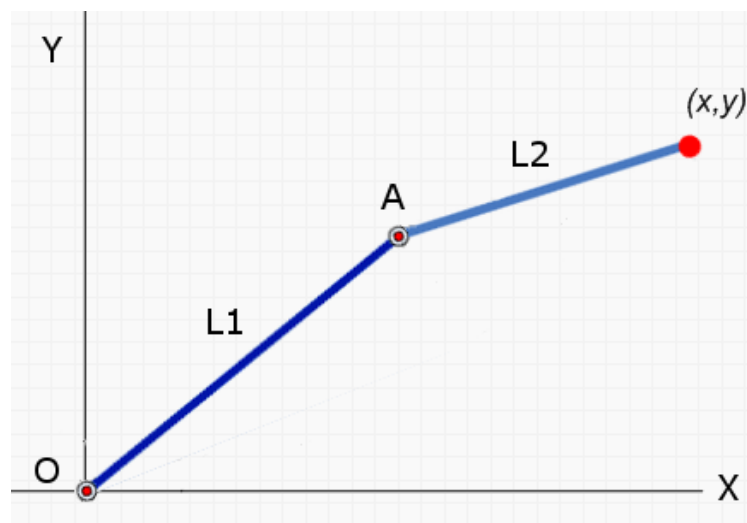


Рис. 4.20. Можливості задання одного і того ж положення робочого органу (x, y) при різних кутах θ_1 і θ_2

Формули для θ_1 і θ_2 не зміняться, але зміняться знаки кутів:

$$\theta_1 = q_1 + q_2. \quad (4.17)$$

Але θ_2 потрібно брати із протилежним знаком [76].

4.6. Питання та завдання для самоперевірки

1. Які стилі проектування мехатронних систем вам відомі?
2. Розкрийте сутність основних положень, покладених в основу проектування мехатронних об'єктів?
3. Поясніть алгоритм проектування мехатронних модулів.
4. Які етапи проектування мехатронної системи із мікропроцесорними пристроями?
5. В чому різниця між мехатронними виконавчими модулями першого та другого покоління?
6. Які рівні програмування мікроконтролерів вам відомі?
7. Що таке структурне програмування?
8. Які програмні засоби використовуються для розробки та налаштування програмного забезпечення мікроконтролерів?
9. Що таке нанотехнології та яка їх роль в мехатроніці?
10. За якими показниками класифікують промислових роботів?
11. Як показники покладені в основу класифікації методів керування роботами?
12. Поясніть основні функції робота.
13. Які системи координат використовують сучасні маніпуляційні системи роботів?
14. За якими ознаками відбувається розпізнавання об'єктів з якими працює робот?
15. Розкрийте сутність програмного та адаптивного керування робототехнічними системами.
16. Розв'яжіть пряму задачу кінематики дволанкового маніпулятора зображеного на рис. 4.18 при наступних параметрах: $L_1=0,4$ м, $L_2=0,3$ м, $\theta_1=0,4$ рад, $\theta_2=0,1$ рад.
17. Розв'яжіть зворотну задачу кінематики дволанкового маніпулятора (рис. 4.18) при наступних параметрах: $L_1=0,4$ м, $L_2=0,3$ м, $x=0,8$ м, $y=0,5$ м.

ДОДАТОК 1

Приклади розв'язування задач оптимального керування

Постановка задачі

Для більшості мехатронних машин робочі органи у процесі руху на початковій стадії моделювання (у першому наближенні) можна представити у вигляді руху матеріальної точки. Диференціальне рівняння руху одномасової механічної системи має такий вигляд:

$$m\ddot{x} = F - W, \quad (1)$$

де m - приведена до поступального руху маса системи; x - узагальнена координата системи (поступальне переміщення); F - приводне зусилля; W - сила статичного опору руху системи.

Крапка над символом означає диференціювання за часом.

Рівняння (1), з урахуванням позначень $x = x_1$, $\frac{F - W}{m} = u$ можна подати у канонічному вигляді:

$$\begin{cases} \dot{x}_1 = x_2, \\ \dot{x}_2 = u. \end{cases} \quad (2)$$

Виконаємо синтез оптимального керування для системи, рух якої описується рівняннями (1). При цьому критерієм оптимізації оберемо інтегральний функціонал, який за структурою є комплексним критерієм:

$$I_{\text{єтт}} = \int_0^T (k_1 x_1^2 + k_2 x_2^2 + k_3 u^2) dt, \quad (3)$$

де T – тривалість руху системи; k_1 , k_2 , k_3 - коефіцієнти при відповідних доданках у підінтегральному виразі, причому:

$$\begin{cases} k_1 = \delta_1 \tilde{I}_1^{-1}, \\ k_2 = \delta_2 \tilde{I}_2^{-1}, \\ k_3 = (1 - \delta_1 - \delta_2) \tilde{I}_3^{-1}. \end{cases} \quad (4)$$

де δ_1 та δ_2 - вагові коефіцієнти, які враховують важливість першого та другого доданків у підінтегральному виразі критерію (3); \tilde{I}_1 , \tilde{I}_2 , \tilde{I}_3 - мінімально можливі значення одиничних критеріїв (наприклад, \tilde{I}_1 є мінімально можливою величиною критерію (3) при умові $\delta_1 = 1$).

Будемо шукати оптимальне керування при таких крайових умовах:

$$\begin{cases} x_1(0) = s_0, x_2(0) = v_0; \\ x_1(T) = x_2(T) = 0. \end{cases} \quad (5)$$

Таким чином, необхідно перевести механічну систему з деякого початкового положення, яке характеризується ненульовими значеннями положення s_0 та швидкості v_0 у нульове положення при мінімізації критерію за виразом (3). Для механічної системи такий режим руху означає загальмовування.

Розв'язування задачі методом динамічного програмування

Для розв'язання цієї задачі використаємо динамічне програмування. Основне функціональне рівняння Беллмана для критерію (3) прийме такий вигляд:

$$0 = \min \left[k_1 x_1^2 + k_2 x_2^2 + k_3 u^2 + \frac{\partial S}{\partial x_1} x_2 + \frac{\partial S}{\partial x_2} u \right], \quad (6)$$

де S - функція Беллмана.

Мінімум правої частини рівняння (6) будемо шукати по параметру керування u для чого продиференціюємо її за u та прирівняємо отримане до нуля:

$$2k_3 u + \frac{\partial S}{\partial x_2} = 0. \quad (7)$$

Знайдемо з рівняння (7) керування u :

$$u = -\frac{1}{2k_3} \frac{\partial S}{\partial x_2} \quad (Д8)$$

та підставимо отримане у рівняння (6), в результаті чого будемо мати:

$$k_1 x_1^2 + k_2 x_2^2 + \frac{\partial S}{\partial x_1} x_2 - \frac{1}{4k_3} \left(\frac{\partial S}{\partial x_2} \right)^2 = 0. \quad (Д9)$$

Рівняння (9) є нелінійним диференціальним рівнянням у частинних похідних. Будемо шукати його розв'язок у вигляді квадратичної форми, яка задовольняє кінцеві умови руху системи (5):

$$S = A_1 x_1^2 + A_2 x_1 x_2 + A_3 x_2^2 \quad (10)$$

Візьмемо частинні похідні з виразу (10) за параметрами x_1 та x_2 :

$$\frac{\partial S}{\partial x_1} = 2A_1 x_1 + A_2 x_2, \quad (11)$$

$$\frac{\partial S}{\partial x_2} = A_2 x_1 + 2A_3 x_2. \quad (12)$$

Підставимо вирази (11) і (12) у рівняння (9) і отримаємо:

$$x_1^2 \left(k_1 - \frac{A_2^2}{4k_3} \right) + x_2^2 \left(k_2 + A_2 - \frac{A_3^2}{k_3} \right) + x_1 x_2 \left(2A_1 - \frac{A_2 A_3}{k_3} \right) = 0. \quad (13)$$

Рівняння (13) буде справедливим у тому випадку коли вирази у дужках будуть рівні нулю, оскільки $x_1 \neq 0$, $x_2 \neq 0$. Тому рівняння (13) можна замінити на систему нелінійних алгебраїчних рівнянь:

$$\begin{cases} k_1 - \frac{A_2^2}{4k_3} = 0, \\ k_2 + A_2 - \frac{A_3^2}{k_3} = 0, \\ 2A_1 - \frac{A_2 A_3}{k_3} = 0. \end{cases} \quad (14)$$

Розв'язок системи рівнянь (14) буде мати два дійсних та два комплексних кореня. Оберемо один дійсний, який не приводить до втрати стійкості системи, оскільки рух системи при цьому є плавним, а максимальна величина керування – незначною.

Підставивши отримані корені у вираз (10) та враховуючи рівність (8), отримаємо функцію оптимального керування у вигляді зворотного зв'язку:

$$u = - \frac{\sqrt{k_1} \sqrt{k_3} x_1 + \sqrt{(k_2 + 2\sqrt{k_1} \sqrt{k_3}) k_3} x_2}{k_3}. \quad (15)$$

Крім того, знайдемо функції для фазових координат системи. Для цього підставимо керування у диференціальне рівняння (1) та розв'яжемо його при відповідних початкових умовах (5). У результаті отримаємо:

$$\begin{aligned} x_1 = & \left[\sqrt{-k_2 + 2\sqrt{k_1} \sqrt{k_3}} s_0 \cos\left(\frac{\sqrt{-k_2 + 2\sqrt{k_1} \sqrt{k_3}} t}{2\sqrt{k_3}}\right) + (2k_3 v_0 + \sqrt{(k_2 + 2\sqrt{k_1} \sqrt{k_3}) k_3} s_0) \times \right. \\ & \times \sin\left(\frac{\sqrt{-k_2 + 2\sqrt{k_1} \sqrt{k_3}} t}{2\sqrt{k_3}}\right) \left. \right] \left\{ \cosh\left(\frac{\sqrt{k_2 + 2\sqrt{k_1} \sqrt{k_3}} t}{2\sqrt{k_3}}\right) - \sinh\left(\frac{\sqrt{k_2 + 2\sqrt{k_1} \sqrt{k_3}} t}{2\sqrt{k_3}}\right) \right\} \times \\ & \times [\sqrt{-k_2 + 2\sqrt{k_1} \sqrt{k_3}} \sqrt{k_3}]^{-1}; \end{aligned} \quad (16)$$

$$\begin{aligned} x_2 = & - \left[-\sqrt{-k_2 + 2\sqrt{k_1} \sqrt{k_3}} \sqrt{k_3} v_0 \cos\left(\frac{\sqrt{-k_2 + 2\sqrt{k_1} \sqrt{k_3}} t}{2\sqrt{k_3}}\right) + (\sqrt{(k_2 + 2\sqrt{k_1} \sqrt{k_3}) k_3} \times \right. \\ & \times v_0 + 2\sqrt{k_1} \sqrt{k_3} s_0) \sin\left(\frac{\sqrt{-k_2 + 2\sqrt{k_1} \sqrt{k_3}} t}{2\sqrt{k_3}}\right) \left. \right] \left\{ \sqrt{(k_2 + 2\sqrt{k_1} \sqrt{k_3}) k_3} \times \right. \\ & \times \cosh\left(\frac{\sqrt{-k_2 + 2\sqrt{k_1} \sqrt{k_3}} t}{2\sqrt{k_3}}\right) - \sqrt{k_2 + 2\sqrt{k_1} \sqrt{k_3}} \sqrt{k_3} \sinh\left(\frac{\sqrt{-k_2 + 2\sqrt{k_1} \sqrt{k_3}} t}{2\sqrt{k_3}}\right) \left. \right\} \times \\ & \times [\sqrt{-k_2 + 2\sqrt{k_1} \sqrt{k_3}} \sqrt{k_3} \sqrt{(k_2 + 2\sqrt{k_1} \sqrt{k_3}) k_3}]^{-1}. \end{aligned} \quad (17)$$

Зазначимо, що такі функції переміщення та швидкості отримано при умові, що на систему не діють стохастичні збурення. Якщо ж такі збурення будуть присутні, то x_1 , x_2 та u будуть відрізнятися від приведених, але вони будуть оптимальними. Для практичної реалізації керування у вигляді зворотного зв'язку необхідно мати датчики положення та швидкості механічної системи. Формування керуючої величини у конкретний момент часу повинен базуватись на вимірах цих датчиків. Таким чином, керуюча функція наперед невідома, вона синтезується протягом руху системи за допомогою механізму зворотного зв'язку.

Для приведених функцій побудуємо графіки та фазову траєкторію (рис. 1).

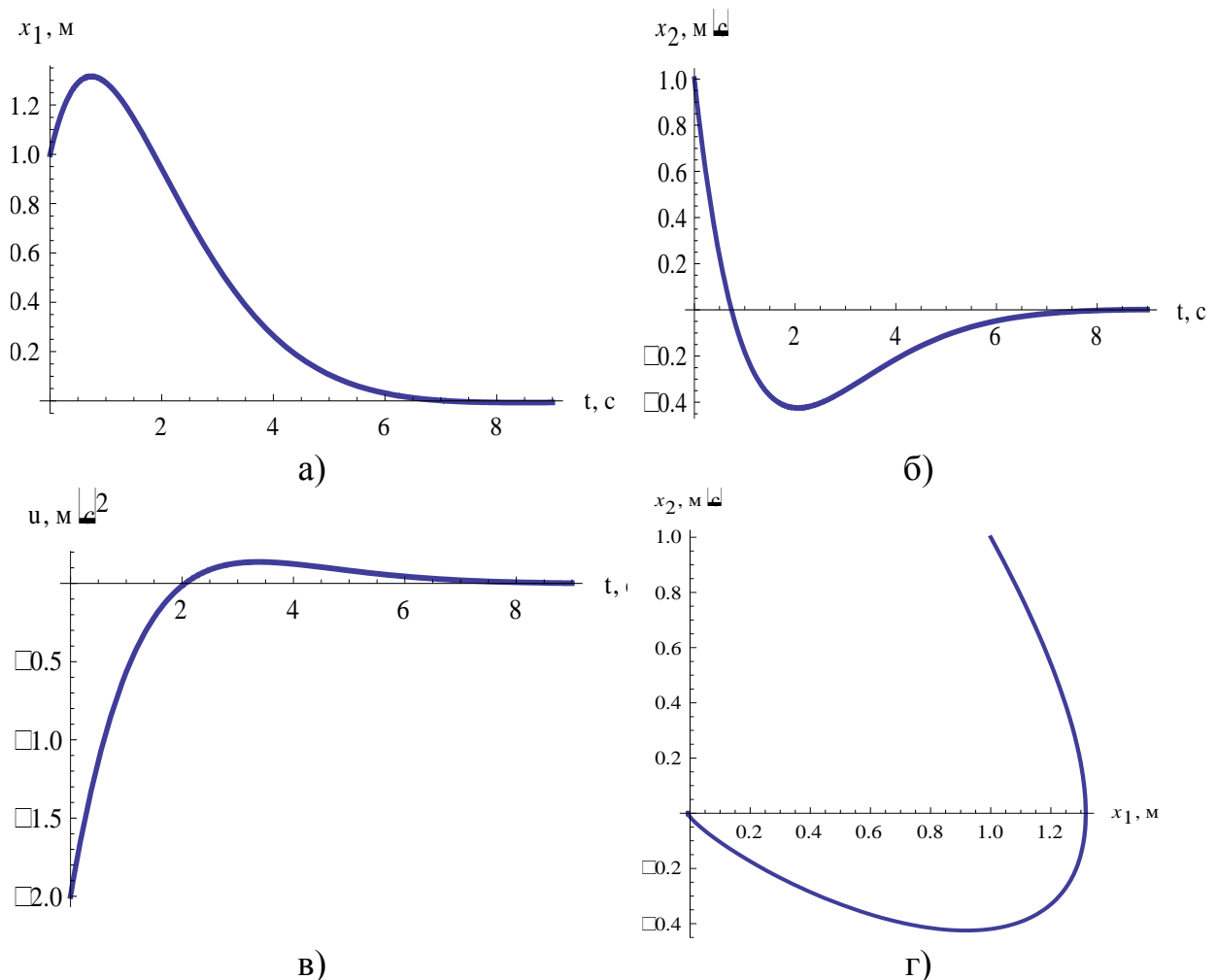


Рис. 1. Графіки функцій фазових координат системи для керування у вигляді зворотного зв'язку: а) переміщення; б) швидкість; в) прискорення (функція керування); г) фазова траєкторія

Використання класичного варіаційного числення

Розв'яжемо оптимізаційну задачу (1), (3), (5) класичним варіаційним методом. Умовою мінімуму функціоналу за виразом (ДЗ) є рівняння Ейлера-Пуассона:

$$x - \frac{k_2}{k_3} \ddot{x} + \frac{k_1}{k_3} x = 0. \quad (18)$$

Для того, щоб розв'язати це однорідне диференціальне рівняння необхідно скласти відповідне характеристичне рівняння:

$$p^4 - \frac{k_2}{k_3} p^2 + \frac{k_1}{k_3} p = 0. \quad (19)$$

Розв'язок рівняння (19) має два дійсних (один з яких рівний нулю $p_1=0$) та два комплексних спряжених кореня. Тому розв'язок рівняння (19) необхідно шукати у вигляді:

$$x = C_1 + C_2 e^{p_2 t} + e^{t\alpha} [C_3 \sin(t\beta) + C_4 \cos(t\beta)], \quad (20)$$

де C_1, C_2, C_3, C_4 - постійні інтегрування, які визначаються за допомогою крайових умов (5); p_2 - дійсний ненульовий корінь рівняння (19); α, β - дійсна та комплексна частина спряжених комплексних коренів рівняння (19), тобто $p_{3,4} = \alpha \pm i\beta$.

Не будемо приводити розв'язок крайової задачі (18), (5) оскільки він має значний об'єм. Приведемо графічну інтерпретацію отриманих функціональних залежностей. Для оптимального керування оберемо тривалість режиму руху меншу ніж для оптимального керування у вигляді зворотного зв'язку. Зазначимо, що неможливо наперед визначити тривалість руху для керування у вигляді зворотного зв'язку, тому задамо деяку апіорну величину тривалості руху. На рис. 2 приведені графіки для обох керувань та відповідні їм графіки функцій фазових координат системи. Графіки для

оптимального керування, яке відповідає задачі (Д18), (Д5) зображено тоншою лінією.

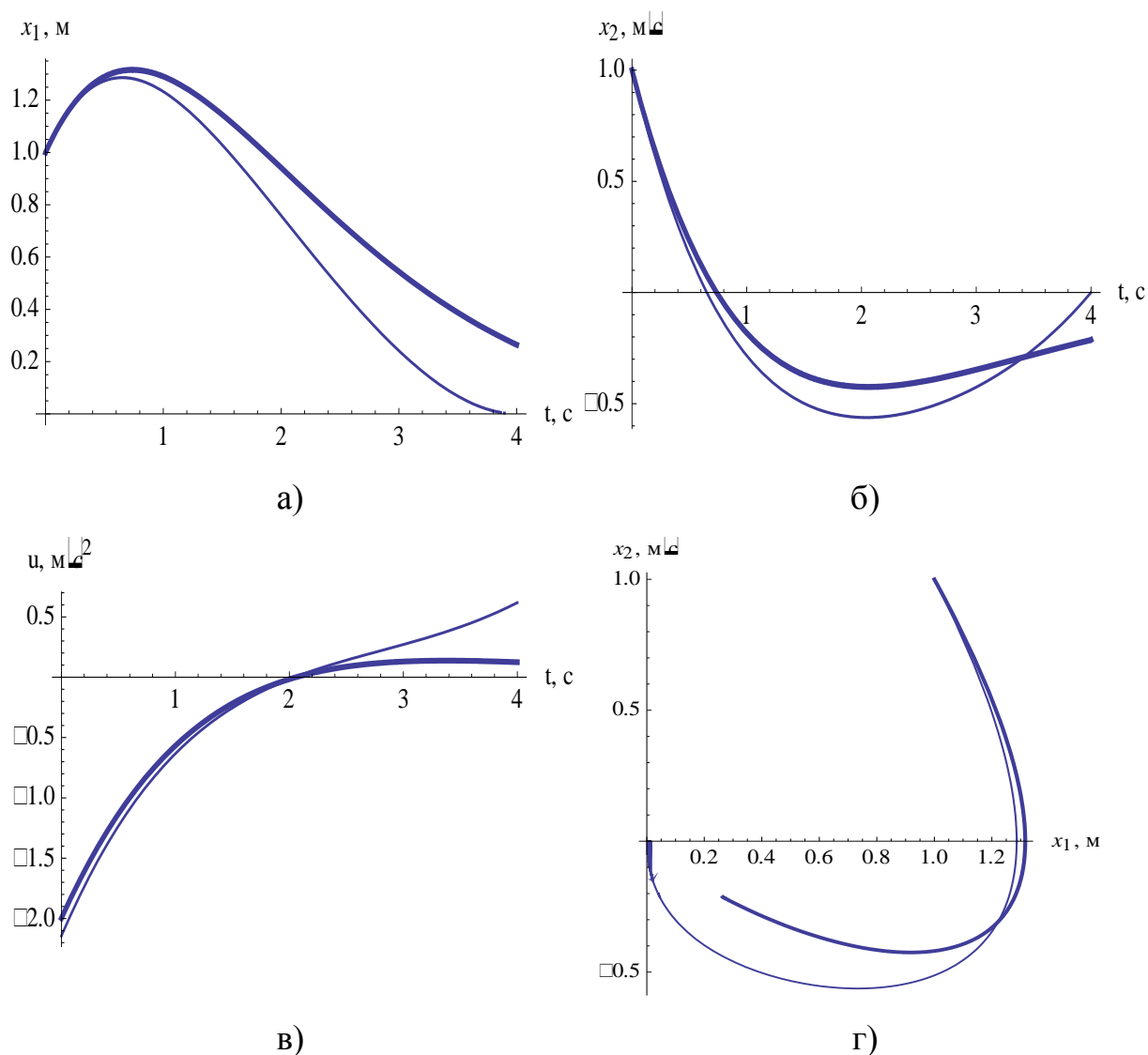


Рис. 2. Графіки функцій фазових координат системи для оптимальних керувань: а) переміщення; б) швидкість; в) прискорення (функція керування); г) фазова траєкторія

З графіків приведених на рис. Д2. видно, що тривалість $T=4$ с недостатня для того, щоб реалізувати керування у вигляді зворотного зв'язку: система не встигає перейти у нульовий стан. Програмне оптимальне керування дозволяє перейти у нульовий стан за будь який час T , що є перевагою такого виду керування, оскільки наперед відомо тривалість руху системи і її можна регулювати. Це необхідно для того, щоб узгодити між

собою складні рухи складних механічних систем (наприклад, крани-штабелери, промислові роботи тощо). Для того, щоб визначити тривалість руху системи для керування у вигляді зворотного зв'язку необхідно розв'язати систему нелінійних алгебраїчних рівнянь, що є складною задачею. Можна шукати розв'язок чисельними методами, або знаходити розв'язок для конкретних умов руху при реалізації керування на практиці.

Розв'язання оптимізаційної задачі методом принципу максимуму

Для того, щоб розв'язати поставлену задачу методом принципу максимуму необхідно записати функцію Гамільтона:

$$\dot{I} = \psi_1 \dot{\delta}_2 + \psi_2 u - k_1 x_1^2 - k_2 x_2^2 - k_3 u^2, \quad (21)$$

де ψ_1 і ψ_2 - спряжені змінні. Згідно принципу максимуму необхідно таким чином керувати процесом, щоб Гамільтоніан (21) був максимальним. Для відкритої області керування ($u \in (-\infty; \infty)$) таке керування знаходиться з умов стаціонарності функції Гамільтона:

$$\frac{\partial \dot{I}}{\partial u} = \psi_2 - 2k_3 u = 0. \quad (22)$$

З рівняння (22) отримаємо:

$$u = \frac{\psi_2}{2k_3}. \quad (Д23)$$

Для того, щоб пересвідчитись, що отримане керування (23) доставляє максимум Гамільтоніану (21), необхідно проаналізувати знак другої похідної Гамільтоніана по керуванню:

$$\frac{\partial^2 \dot{I}}{\partial u^2} = -2k_3 < 0, \quad (k_3 > 0). \quad (24)$$

Вираз (24) справедливий для відкритої області керування. Однак, як правило, на керування накладаються обмеження у вигляді нестрогих нерівностей:

$$u_{\max} \geq u \geq u_{\min}, \quad (25)$$

де u_{\max} та u_{\min} - відповідно максимальне та мінімальне значення керування. Тоді область допустимих керувань буде обмежена границями u_{\max} та u_{\min} . Враховуючи обмеження (25) оптимальне керування можна подати у такому вигляді:

$$u = \begin{cases} u_{\max}, & \text{якщо } \frac{\psi_2}{2k_3} \geq u_{\max}; \\ \frac{\psi_2}{2k_3}, & \text{якщо } u_{\max} \geq \frac{\psi_2}{2k_3} \geq u_{\min}; \\ u_{\min}, & \text{якщо } \frac{\psi_2}{2k_3} \leq u_{\min}. \end{cases} \quad (26)$$

Вираз (26) не дає відповіді про те як керувати мехатронною системою. Це лише якісна картина керування. Для знаходження кількісних параметрів, наприклад, тривалості окремих періодів сталості керування необхідно проводити додаткове дослідження. Мета цього дослідження - визначити спряжену функцію $\psi_2 = \psi_2(t)$ (програмне керування) або $\psi_2 = \psi_2(x_1, x_2)$ (керування у вигляді зворотного зв'язку).

Зв'язок між принципом максимуму та класичним варіаційним численням

Для знаходження невідомої спряженої змінної ψ_2 необхідно знайти розв'язки спряженої системи диференціальних рівнянь:

$$\begin{cases} \dot{\psi}_1 = -\frac{\partial \dot{I}}{\partial \delta_1} = 2\delta_1 k_1; \\ \dot{\psi}_2 = -\frac{\partial \dot{I}}{\partial \delta_2} = -\psi_1 + 2\delta_2 k_2. \end{cases} \quad (27)$$

Продиференціюємо останнє рівняння системи (27) за часом та враховуючи перше рівняння (27) отримаємо:

$$\ddot{\psi}_2 = -2\tilde{\delta}_1 k_1 + 2uk_2. \quad (28)$$

Надалі двічі продиференціюємо вираз (28) за часом та підставимо у вираз (23). В результаті будемо мати:

$$\ddot{u} = \frac{\ddot{\psi}_2}{2k_3} = \frac{-2\tilde{\delta}_1 k_1 + 2uk_2}{2k_3}. \quad (29)$$

Враховуючи, що $\ddot{u} = \overset{IV}{\delta}$ і $u = \ddot{x}$ отримаємо відоме рівняння Ейлера-Пуассона (18).

ДОДАТОК 2

Розробка програмної та апаратної частин системи керування моделлю вантажопідйомного крана

Розробка конструкції моделі вантажопідйомного крана

Модель вантажопідйомного крана являє собою кран-балку (двутавр №20) встановлену на опорах (рис. 1). По полицях кран-балки рухається візок (електрична таль). Привод візка здійснюється від асинхронного короткозамкненого двигуна ФТТ-0,08/4. Живлення приводу здійснюється від кабелів, підвішених на тросі.



Рис. 1. Зовнішній вигляд фізичної моделі крана

Електрична схема фізичної моделі представлена на рис. 2. Для керування електродвигуном приводу моделі кранового візка обрано частотний

перетворювач FR-S540-E-2.2K-EC (Mitsubishi Electric), основні технічні характеристики якого приведені в табл. 1.

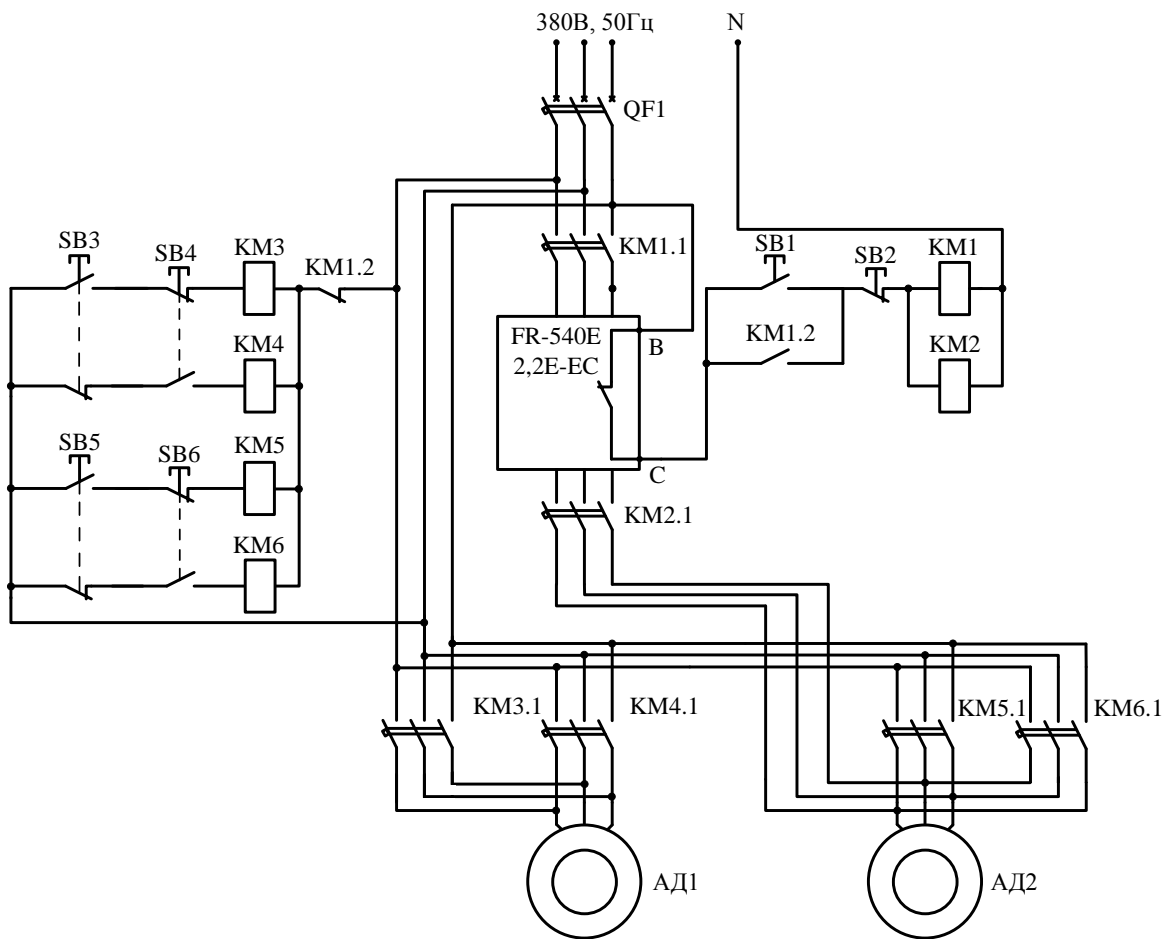


Рис. 2. Принципова електрична схема фізичної моделі.

Таблиця 1. Технічні характеристики частотного перетворювача
FR-S540-E-2.2K-EC

Параметр	Значення параметру
1	2
Потужність перетворювача	2,2 кВт
Номінальна вихідна потужність	3,7 кВА
Потужність джерела живлення	5,5 кВА
Номінальний вихідний струм	4,8 А
Перевантаження	150% при 60 с; 200% при 0,5 с
Напруга живлення	380 В

1		2
Допустимий інтервал напруг		323-528 В
Діапазон вихідних частот		0,5...120 Гц
Допустиме відхилення частоти		±5 %
Час розгону/гальмування		0; 0,1;...999 с
Тип характеристики розгону/гальмування		лінійна, S-подібна
Максимальний пусковий момент		0,15M _{ном}
Способи гальмування	1) генераторне гальмування	гальмівний момент 0,2M _{ном}
	2) динамічне гальмування	час гальмування 0...10 с
		напруга гальмування 0,15U _{ном}
Керування		1) зовнішнє 2) керування з пульта 3) комп'ютерне
Ступінь захисту		IP 20
Охолодження		примусове повітряне

Для зменшення похибки задання вихідної частоти задіяна функція компенсації ковзання (параметри частотного перетворювача P95, P96, P97). Активація цієї функції дозволяє за допомогою вихідного струму перетворювача обчислювати ковзання двигуна і таким чином коригувати поточну вихідну частоту, якщо це потрібно.

Даний частотний перетворювач може реалізовувати закони $\frac{U}{U_{\bar{m}}} = \frac{f}{f_{\bar{m}}}$

та $\frac{U}{U_{\bar{m}}} = \left(\frac{f}{f_{\bar{m}}}\right)^2$. Було встановлено такий закон керування зміни напруги

живлення двигуна при зміні частоти: $\frac{U}{U_{\bar{m}}} = \frac{f}{f_{\bar{m}}}$ (параметр частотного

перетворювача $P14=0$). Вибір цього закону обумовлений тим, що штучні механічні характеристики двигуна за невеликих частот напруги живлення мають достатньо великі критичні моменти. Тому задання цієї характеристики дозволить двигуну візка краще відпрацьовувати задану швидкість.

Фізична модель дозволяє реалізовувати як ручне так і комп'ютерне керування рухом моделі крана. Ручне керування здійснюється за допомогою кнопочного поста (кнопки SB3...SB6). Кнопками SB3 або SB4 вмикається двигун горизонтального переміщення моделі АД1, кнопками SB5 або SB6 – двигун вертикального переміщення вантажу АД2.

Перемикання режимів керування здійснюється за допомогою кнопки SB1. При цьому розмикається контакт КМ1.2 та вимикає коло живлення кнопочного поста, також замикаються контакти пускачів КМ1 та КМ2 і підключають частотний перетворювач FR-540E-2,2K-EC до клем двигуна АД1. Отже з'являється можливість живити двигун струмом з частотою, яка буде змінюватись за певним законом.

Коло перемикання режимів з'єднано через нормально замкнений контакт частотного перетворювач (клеми В, С). Це зроблено для того, щоб унеможливити виникнення аварійних станів перетворювача: якщо такі стани будуть виникати (перевантаження по струму, перегрів, перенапруження тощо), то контакт розімкнеться та знеструмить коло керування, що в свою чергу автоматично відключить перетворювач та переведе керування моделі в ручний режим.

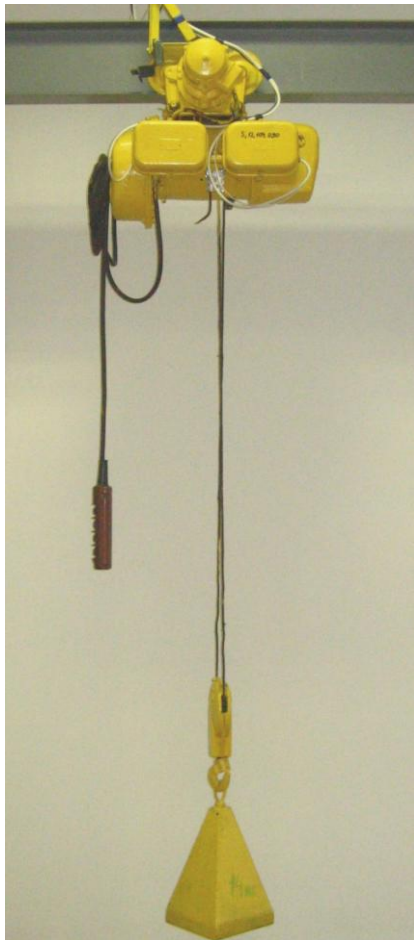
Ручне перемикання з автоматичного в ручний режим здійснюється за допомогою кнопки SB2. Вмикання схеми здійснюється за допомогою автоматичного вимикача QF1.

Все стаціонарне електрообладнання (частотний перетворювач, пускачі КМ1 та КМ2, кнопки SB1 та SB2) змонтовано в щитку (рис. 3).



Рис. 3. Щиток з електрообладнанням для автоматичного керування рухом візка

Електрообладнання, яке здійснює ручне керування (кнопковий пост, пускачі КМЗ...КМ6) змонтовано на візку. До візка підходить два гнучких кабелі живлення: для живлення струмом постійної (4 жили) та змінної (за допомогою частотного перетворювача) частоти (3 жили) (рис. 4).



а)



б)

Рис. 4. Зовнішній вигляд моделі візка (електроталь): а) модель візка з вантажем; б) модель вантажного візка.

Розробка програмного забезпечення для керування рухом моделі крана

Для реалізації керування рухом візка протягом всього циклу горизонтального переміщення використовується комп'ютерне керування частотним перетворювачем. Таке керування дає змогу розганяти та гальмувати модель крана за оптимальними законами з врахуванням всіх змінних: довжини переміщення, довжини гнучкого підвісу вантажу, маси транспортованого вантажу, номінальної швидкості моделі крана, її допустимого прискорення та тривалості перехідних процесів руху.

Суть комп'ютерного керування полягає в наступному: з персонального комп'ютера на частотний перетворювач надходять дискретні значення

вихідної частоти напруги живлення двигуна візка. У відповідності з цими сигналами частотний перетворювач змінює вихідну частоту і електродвигун візка змінює швидкість обертання вала, що, в результаті, розганяє або гальмує візок за певними законами руху. Протягом усталеного руху частотний перетворювач підтримує постійну частоту, яка відповідає номінальній швидкості руху.

Структурно програма складається з основного модуля (unit Unit1) та додаткового (функціонального) модуля (unit Functions_Unit), який виконує допоміжні функції та процедури.

Опишемо докладніше процеси формування даних та їх передачу на перетворювач. Для формування масиву дискретних швидкостей використовується спеціально розроблена програма „Оптимальні режими руху”, інтерфейс якої представлено на рис. 5.

Програма працює таким чином: користувач задає значення відстані переміщення візка, довжину гнучкого підвісу вантажу, при якому здійснюється транспортування вантажу, номінальну швидкість руху візка (в процентах від номінальної швидкості при частоті напруги живлення 50 Гц), а також значення допустимого прискорення візка. В програмі задається час розгону візка ($t_1=1$ с). Для заданого часу розгону візка визначається довжина масиву прискорень. При часі дискретизації $\Delta t=0,1$ с і введеному початковому часі розгону візка довжина масиву складе 10 елементів. Далі програма заповнює елементи масиву даними - конкретними значеннями прискорення для конкретних моментів часу, які визначаються з формули для прискорення при попередньо обраному режимі руху. Причому кожен наступний елемент масиву визначається при часі на 0,1 с більше ніж попередній. У кінці цієї процедури отримуємо масив прискорень з яких необхідно визначити найбільше значення.

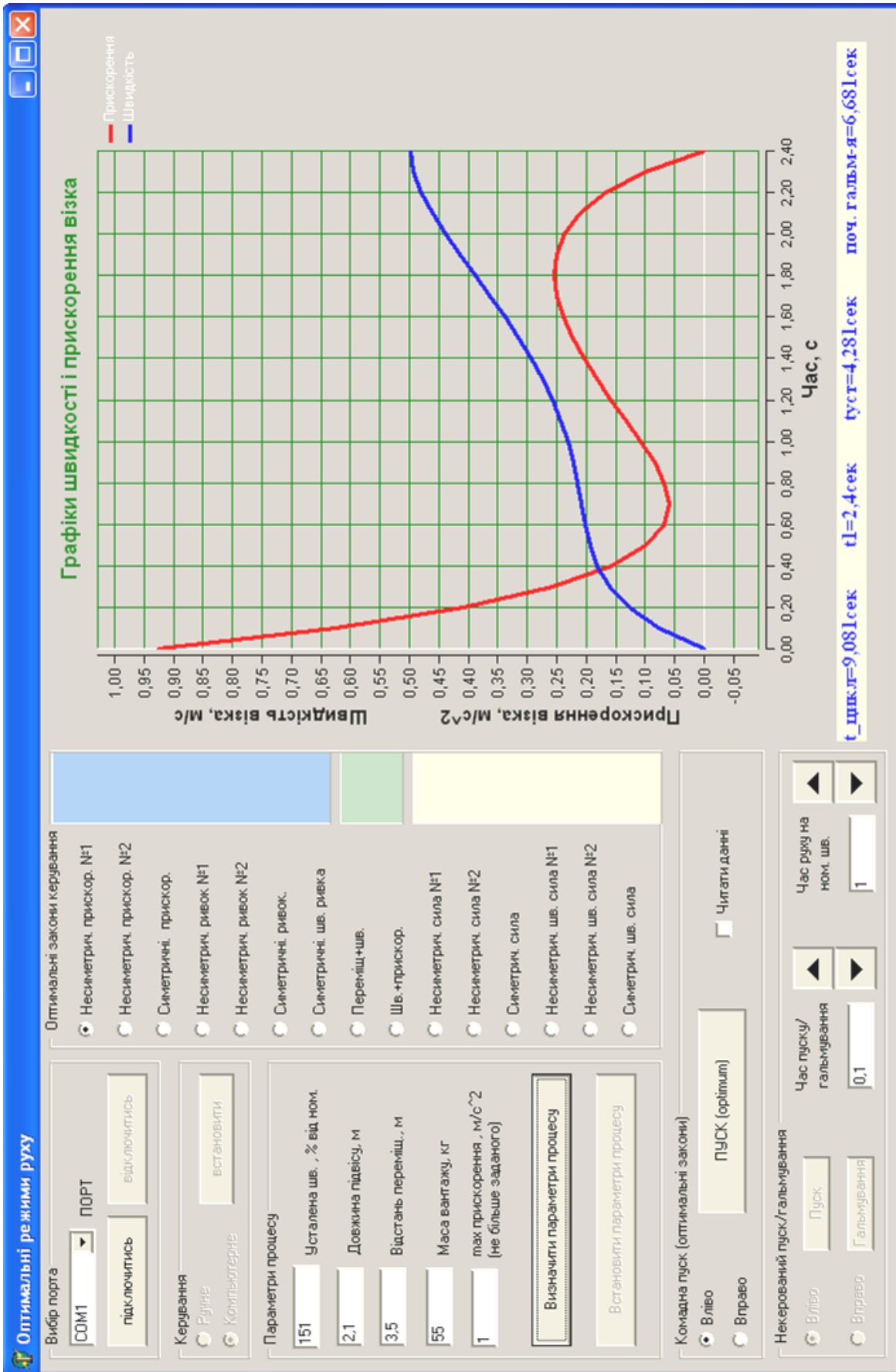


Рис. 5. Зовнішній вигляд вікна розробленої програми

Надалі у програмі порівнюються всі елементи масиву прискорень між собою. Визначене максимальне значення прискорення порівнюється з допустимим, і якщо допустиме прискорення буде менше ніж максимальне, то програма збільшує час розгону на 0,1 с і повторює цикл. Цикли повторюються допоки не виконається умова $\tilde{a}_{1\max} \leq \dot{a}_{\text{дп}}$ (максимальне прискорення візка не більше допустимого). Якщо ж ця умова виконується, то далі програма визначає довжину переміщення візка протягом перехідних режимів руху та порівнює її з величиною заданої відстані переміщення. При невиконанні умови $2\tilde{\sigma}_1(t_1) \leq x$ (задана відстань переміщення повинна бути не меншою ніж сума відстаней, які візок проїде при розгоні та гальмуванні) програма видає повідомлення „Введіть менше значення усталеної швидкості або більшу відстань переміщення” та виходить з циклу. У випадку виконання умови відбувається формування часу усталеного руху візка ($t_{\text{обд}} = \frac{x - 2\tilde{\sigma}_1(t_1)}{\tilde{\sigma}_{1\text{м}}}$).

Довжина масиву прискорень рівна довжині масиву для дискретних значень швидкості. Таким чином, маючи кількість дискретних точок швидкості візка програма обчислює їх (це відбувається аналогічно обчисленню дискретних прискорень) та записує в масив.

Для здійснення підготовки частотного перетворювача до роботи необхідно задати час розгону та гальмування (параметри P7, P8 задано 0,1 с). Задання здійснюється програмно – відправкою на частотний перетворювач цифрових кодованих значень цих параметрів. Необхідно також встановити параметр лінійної характеристики розгону/гальмування (параметр P29 встановлюється в 0). Вибір саме таких параметрів розгону/гальмування обумовлений тим, що за такого керування швидкість руху візка буде представляти собою ламану криву, лінійні ділянки якої будуть сполучатись в точках швидкостей, які визначені раніше. Зі збільшенням часу розгону/гальмування збільшується кількість точок, тому реальна крива буде наближатись до ідеальної краще, але при цьому затягуються перехідні процеси руху візка.

Для унаочнення роботи програми представимо її функціональну блок-схему (рис. 6).

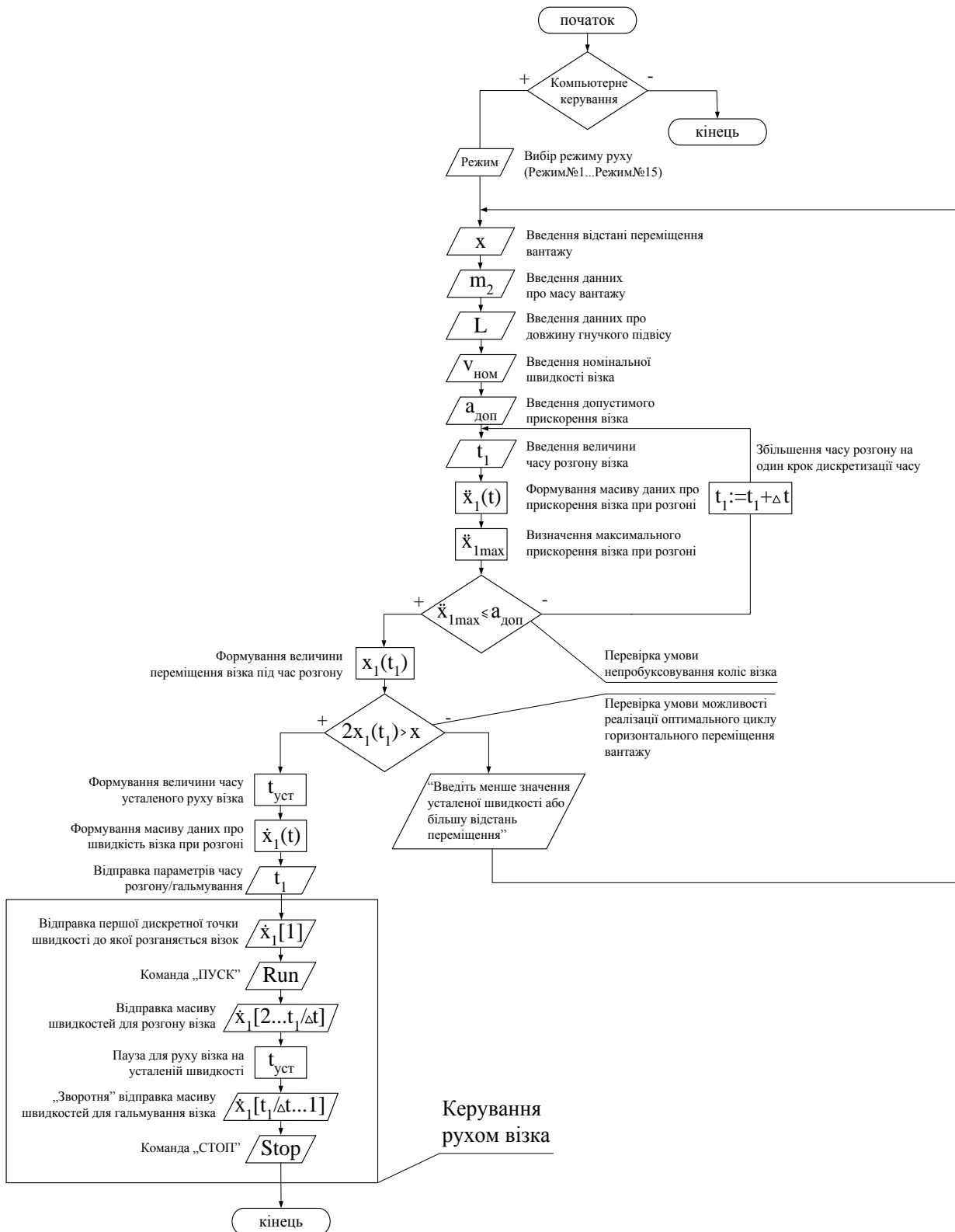


Рис. 5. Функціональна блок-схема роботи розробленої програми „Оптимальні режими руху”.

Керування швидкістю обертання вала двигуна здійснюється шляхом відправки на частотний перетворювач цифрових сигналів, які вказують яку вихідну частоту необхідно „відпрацювати”. Процес відправки масиву швидкостей для всього циклу переміщення візка здійснюється так: спочатку відправляється початкове значення частоти (початкової швидкості руху візка), до якої візок розженеться через 0,1 с, а потім дається команда на пуск. При цьому частотний перетворювач видає початкову частоту і візок починає рух. Через 0,1 с, коли початкова швидкість візка моделі крана буде досягнута, з комп'ютера на частотний перетворювач відправляється наступне кодоване значення частоти. Електродвигун переміщення візка, у відповідності до збільшення вихідної частоти живлення, збільшує кількість обертів і через 0,1 с візок досягає наступного значення дискретної швидкості. Процес повторюється до досягнення візком номінальної швидкості. Далі візок рухається на усталеній швидкості, яка являє собою останню дискретну швидкість з масиву. Під час усталеного руху програма „спить” протягом часу, який визначено раніше (t_{ycm}).

Після закінчення часу t_{ycm} програма відправляє на частотний перетворювач кодовані сигнали з масиву швидкостей аналогічно до процесу розгону, але в зворотному порядку. Так здійснюється процес гальмування візка. Коли значення останньої точки масиву швидкостей (для розгону це перша точка) буде відпрацьовано, тоді комп'ютер відправить на перетворювач команду на гальмування і візок зупиниться.

Технічна реалізація зв'язку частотного перетворювача та ПК здійснюється за допомогою послідовного порту COM. Апаратна частина реалізації зв'язку здійснюється за допомогою послідовного порта RS485, який має частотний перетворювач. Конвертація сигналу у відповідності до протоколу зв'язку RS232 (COM) здійснюється за допомогою конвертора інтерфейсів SC-FRPC фірми Beijer Electronics. З конвертованим сигналом можна працювати за допомогою функцій зчитування та запису даних на

апаратний COM-порт. Параметри послідовного зв'язку між ПК та частотним перетворювачем приведені в таблиці 2.

Пакети даних для різних типів команд (запис параметру, установка вихідної частоти, команда пуск та стоп) представляються у вигляді послідовностей символів в кодуванні ASCII.

Таблиця 2. Специфікація параметрів зв'язку між ПК та частотним перетворювачем

Елемент	Параметр зв'язку перетворювача	Значення
Метод взаємодії	-	Напівдуплекс
Протокол	-	Асинхронний
Номер станції (перетворювача)	n1	0
Швидкість обміну	n2	192 (19200 біт/с)
Кодування символів – ASCII	n3	0 (8 біт)
Кількість стопових бітів	n3	0 (1 стоповий біт)
Контроль парності	n4	0 (відсутній)
Вибір кінцевих символів CR/LF	n11	1 (CR)
Контрольна сума	-	завжди присутня

Для різних типів команд використовується різні формати „слів”: для запису параметру та для установки вихідної частоти використовуються 13-ти байтові пакети з даними які кодуються чотирма байтами (рис. 6); для команд пуск та стоп – 11-ти байтові з двохбайтовими даними (рис. 7).

Крім того, розроблена програма дозволяє здійснювати керування рухом моделі вантажопідйомного крана з заданими параметрами (тривалість часу розгону та гальмування, тривалість руху на номінальній швидкості), але без усунення коливань вантажу. Це дозволяє моделювати режими руху реальних кранів та досліджувати коливання вантажу, які при цьому виникають та продовжуються після закінчення руху візка.

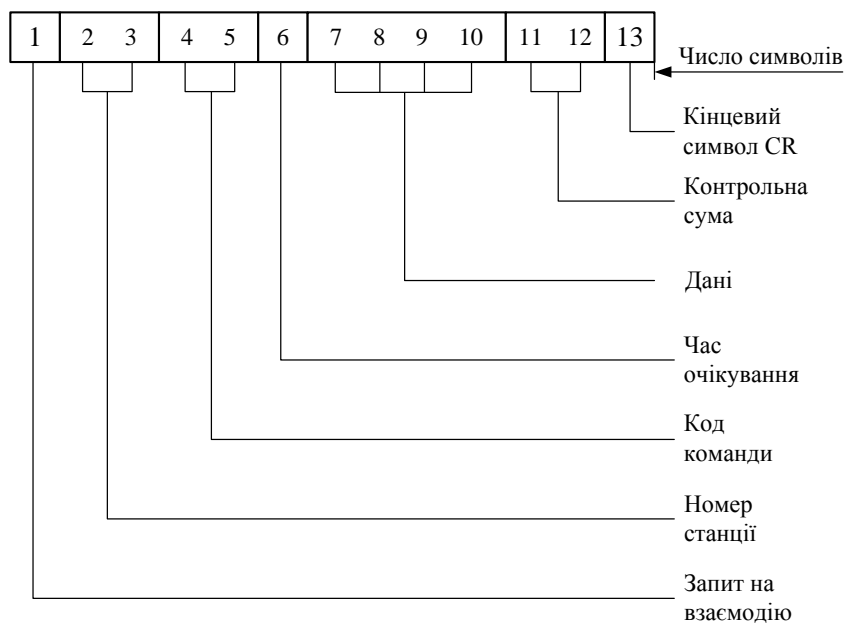


Рис. 6. Формат даних для команд запису параметру та установки вихідної частоти

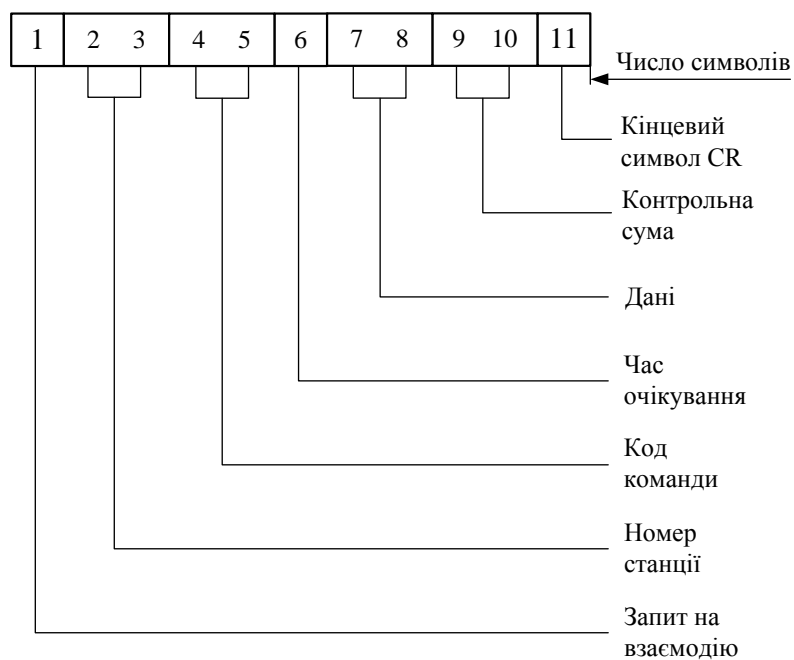


Рис. 7. Формат даних для команд пуск та стоп

Таким чином, керування частотним перетворювачем за допомогою розробленої програми (програмна частина) та послідовного протоколу зв'язку СОМ (апаратна частина) дозволяє здійснювати рух візка з підвішеним до нього вантажем за оптимальними законами.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Harashima F. Mechatronics - what is it, why and how? / F. Harashima, M. Tomizuka, T. Fukuda // IEEE/ASME Transaction on Mechatronics. – vol. 1. – № 1. – 1996. – P. 34-42.
2. Подураев Ю.В. Основы мехатроники [учебное пособие] / Подураев Ю.В. – М.: Изд-во МГТУ „СТАНКИН”, 2000 – 80 с.
3. Оксфордская иллюстрированная энциклопедия в 9-ти томах. Изобретения и технология. - М.: Изд. Дом „Инфра-М”. – Т 6. – 2000. – 185 с.
4. Подураев Ю.В. Принципы построения и современные тенденции развития мехатронных систем / Ю.В. Подураев, В.С. Кулешов // Мехатроника. - 2000. - №1. – С. 5-10.
5. Asar M. Mechatronics: the basis for new industrial development / M. Asar, J. Macra, E. Penney. – Boston: Computational Mechanics Publ. – 1994. – 844 p.
6. Bishop. Robert H. The Mechatronics handbook / Robert H. Bishop. – Austin: The University of Texas at Austin. – 2002. – 1229 p.
7. Мехатроника / [Исин Т., Симояна И., Иноуэ Х. и др.]; под. ред. В.В. Василькова.; пер. с японского С.Л. Масленников. – М.: Мир, 1988. – 318 с.
8. Федеральный государственный образовательный стандарт высшего профессионального образования Российской Федерации. - [Утвержден от 8 декабря 2009 г. приказ № 702]. – 29 с.
9. Осипов Ю.М. Основы мехатроники / Ю.М. Осипов, П.К. Васенин, Д.А. Медведев, С.В. Негодяев. – Томск: Изд-во ТУСУРа, 2007. – 203 с.
10. Алексеев В.О. Информационный анализ и синтез мехатронных систем / В.О. Алексеев // Вестник ХГАДТУ. – №12-13. – 2000. – С. 199-201.
11. Попов Е.П. Основы робототехники / Е.П. Попов, Г.В. Письменный. – М.: Высшая школа, 1990. – 224 с.

12. Горитов А.Н. Моделирование адаптивных мехатронных систем / А.Н. Горитов, А.М. Кориков. – Томск: В-Спектр, 2007. – 292 с.
13. Головки В.М. Теоретичні основи автоматики: Курс лекцій / В.М. Головки. – Ніжин. – 2004. – 104 с.
14. Шандров Б. В. Технические средства автоматизации : учебник для студентов высших учебных заведений / Б. В. Шандров, А. Д. Чудаков. - М.: Издательский центр „Академия”, 2007. — 368 с.
15. Методы практического конструирования при нормировании сигналов с датчиков: по материалам семинара „Practical design techniques for sensor signal conditioning”. [автор перевода Горшков Б.Л.] / Analog Devices, AUTEX Ltd. – С.-Пб.: АВТЭКС, 2005. – 311 с.
16. Хадлстон К. Проектирование интеллектуальных датчиков с помощью Microchip dsPIC / К. Хадлстон. – К.: „МК-Пресс”, 2008. – 320 с.
17. Чиликин М.Г. Общий курс электропривода: учебник для вузов / М.Г. Чиликин, А.С. Сандлер. М.: Энергоиздат, 1981. – 567 с.
18. Фираго Б.И. Теория электропривода: учебное пособие / Б.И. Фираго, Л.Б. Павлячик. – Мн: ЗАО „Техноперспектива”, 2004. – 527 с.
19. Овчинников И.Е. Вентильные электрические двигатели и привод на их основе (малая и средняя мощность): курс лекций / И.Е. Овчинников. – СПб.: КОРОНА-Век, 2006. – 336 с.
20. Костенко М.П. Электрические машины / М.П. Костенко, Л.М. Пиотровский. - М.-Л.: Энергия, 1965. - Ч.2. - 704 с.
21. Штейнер Р.Т. Математическое моделирование электроприводов переменного тока с полупроводниковыми преобразователями частоты / Р.Т Штейнер. - Екатеринбург: УРО РАН, 2000. – 654 с.
22. Терехов В.М., Осипов О.И. Системы управления электроприводов: учебник / Терехов В.М., Осипов О.И.; под ред. Терехова В.М. – Саратов: Изд. центр „Академия”, 2005. – 300 с.

23. Рудаков В.В. Асинхронные электроприводы с векторным управлением / В.В. Рудаков, И.М. Столяров, В.А. Дартау– Л.: Энергоатомиздат, 1987. – 136 с.
24. Соколовский Г.Г. Электроприводы переменного тока с частотным регулированием [учебник] / Г.Г. Соколовский. – М.: Academia, 2006. – 265 с.
25. Пат. 3824437 Germany, Method for controlling asynchronous machines / Felix Blaschke; Siemens Aktiengesellschaft. - № 237572; заявл. 16.07.1971.
26. Усольцев А.А. Частотное управление асинхронными двигателями. Учебное пособие по дисциплинам электромеханического цикла / А.А. Усольцев. – СПб: СПбГУ ИТМО, 2006. – 94 с.
27. Электронный ресурс <http://www.ges.ru/raznoe/triol/29.htm>
28. Лавріненко Ю.М., Марченко О.С., Савченко П.І., Синявський О.Ю., Войтюк Д.Г. Лисенко В.П. Электропривод: підручник (за ред. Лавріненка Ю.М.). – К.: вид-во Лір-К., 2009. – 504 с.
29. Базовые функции преобразователей частоты: за что платим? // Конструктор машиностроитель. 2009. № 1.- С 8-10. (матеріал підготовлений компанією АДЛ).
30. Егоров О.Д. Мехатронные модули. Расчет и конструирование: учебное пособие / О.Д. Егоров, Ю.В. Подураев. – М.: МГТУ „СТАНКИН”, 2004. – 360 с.
31. Денисенко В.В. Компьютерное управление технологическим процессом, экспериментом, оборудованием / В.В. Денисенко. – М.: Горячая линия-Телеком, 2009. – 608 с.
32. Saridis G.N. Journal Automatica (Journal of IFAC) / G.N. Saridis. - Vol. 25. – Issue 3. – 1989. – P. 461 – 467.
33. Интеллектуальные системы автоматического управления / Под ред. И. М. Макарова, В. М. Лохина. – М.: ФИЗМАТЛИТ, 2001. – 576 с.

34. Борисевич А. Теория автоматического управления: элементарное введение с применением MATLAB / А. Борисевич. - СПб.: Изд-во Политехн. ун-та, 2011. - 200 с.
35. Олейников В.А. Основы оптимального и экстремального управления / В.А. Олейников. – М.: Высшая школа, 1969. – 296 с.
35. Іванов А.О. Теорія автоматичного керування / А.О. Іванов. – Дніпропетровськ: Національний гірничий університет. – 2003. – 250 с.
37. Ang K.H. PID control system analysis, design, and technology / K.H. Ang, G. Chong, Y. Li // IEEE Trans. on Control Systems Technology. – Vol. 13. – №. 4. – 2005. – P. 559-576.
38. Власенко В.А. Динамическая настройка стандартных регуляторов / В.А. Власенко, О.К. Мансурова. – СПб: СПбГИТМО (ТУ), 2002. – 52 с
39. Денисенко В. ПИД-регуляторы: вопросы реализации. Часть 1 / В. Денисенко // В записную книжку инженера. - №4. – 2007. – С. 86-97.
40. Денисенко В. ПИД-регуляторы: вопросы реализации. Часть 2 / В. Денисенко // В записную книжку инженера. - №1. – 2008. – С. 86-99.
41. Ловейкин В. С. Расчеты оптимальных режимов движения механизмов строительных машин / Ловейкин В. С. – К.: УМК ВО, 1990. – 168 с.
42. Понтрягин Л.С. Математическая теория оптимальных процессов / Л.С. Понтрягин, В.Г. Болтнянский, Р.В. Гамкрелидзе, Е.Ф. Мищенко. – М.: Физматгиз, 1961. – 392 с.
43. Беллман Р. Динамическое программирование / Р. Беллман. – под. ред. Воробьева Н.Н. – М.: Издательство иностранной литературы, 1960. – 400 с.
44. Петров Ю.П. Вариационные методы теории оптимального управления / Ю.П. Петров. – Л.: Энергия, 1977. – 280 с.
45. Рыбалев А.Н. Теория автоматического управления. Оптимальные системы. Теоретические сведения с примерами решения задач и задания к практическим и лабораторным работам / А.Н. Рыбалев. – Благовещенск: Амурский гос. ун-т, 2006. – 107 с.

46. Федоренко Р.П. Приближенное решение задач оптимального управления / Р.П. Федоренко – М.: Наука, 1978. – 488 с.
47. Zade L.A. The concept of a linguistic variable and its application to approximate reasoning. Part 1 / L.A. Zade // Information Sciences. – № 8. – P. 199-249.
48. Zade L.A. The concept of a linguistic variable and its application to approximate reasoning. Part 2 / L.A. Zade // Information Sciences. – № 8. – P. 301-357.
49. Zade L.A. The concept of a linguistic variable and its application to approximate reasoning. Part 3 / L.A. Zade // Information Sciences. – № 9. – P. 43-80.
50. Гриняев С. Нечеткая логика в системах управления / Гриняев С. // Компьютерра. – 2001. – [Электронный ресурс]. <http://www.computerra.ru/offline/2001/415/13052/>
51. Комашинский В.И. Нейронные сети и их применение в системах управления и связи / В.И. Комашинский, Д.А. Смирнов. – М.: Горячая линия-Телеком, 2003. – 94 с.
52. Каляев И.А. Децентрализованные системы компьютерного управления / И.А. Каляев, Э.В. Мельник. – Ростов н/Д.: Издательство ЮНЦ РАН, 2011. – 196 с.
53. Cucej Z. Industrial networks / Z. Cucej, D. Gleich, M. Kaiser, P. Planinsic // Electronics in Marine.Proceedings 46-th International Symposium Elmar. 16-18 June 2004. – P. 59–66.
54. Thomesse J.-P. Fieldbus Technology in Industrial Automation / J.-P. Thomesse // Proc. of the IEEE. – Vol. 93 – 2005. – P. 1073-1101.
55. Xi B. Use of Ethernet for Industrial Control Networks / B. Xi, Y. Fang, M. Chen, J. Liu // ICIEA – 1st IEEE Conference on Industrial Electronics and Applications. – May 2006. – P. 1-4.
56. Олифер В.Г. Основы сетей передачи данных / В.Г. Олифер, Н.А. Олифер – М.: ИНТУ, 2003 – 248 с.

57. Brooks T. Wireless technology for industrial sensor and control networks / T. Brooks // Proceedings of the First ISA/IEEE Conference Sensor for Industry. – 2001. – P. 73–77.
58. Specification of the Bluetooth System. Master Table of Contents & Compliance Requirements. Covered Core Package version: 2.0 + EDR Current Master TOC issued: 4 November 2004. – 1230 p.
59. ANSI/IEEE Std 802.11. 1999 Edition. Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. – IEEE-SA Standards Board. 12 June 2003. – 528 p.
60. ZigBee specification. Document 053474r13. – ZigBee Standards Organization. Dec. 1. 2006. – 534 p.
61. Willig A. Wireless technology in industrial networks / A. Willig, K. Matheus., A. Wolisz. // Proc. of the IEEE. – Vol. 93. – Issue 6. – 2005. – P. 1130–1151.
62. Akyildiz I.F. A survey on wireless mesh networks / I.F. Akyildiz, X. Wang // IEEE Communications Magazine. – Vol. 43. – Issue 9. – 2005. – P. 223–230.
63. Wiberg P.-A. Wireless technology in industry – Applications and user scenarios / P.-A. Wiberg, U. Bilstrup // Proc. IEEE Int. Conf. Emerging Technologies and Factory Automation (ETFA '01). – P. 123–133.
64. Рошан П. Основы построения беспроводных локальных сетей стандарта 802.11 / П. Рошан, Дж Лиэри. – М.: Издательский дом „Вильямс”, 2004. – 304 с.
65. Генри С. Уоррен мл. Алгоритмические трюки для программистов / Г.С. Уоррен мл. – М.: Издательский дом „Вильямс”, 2007. – 288 с.
66. Готлиб Б.М. Введение в мехатронику: учебное пособие / Б.М. Готлиб. – Екатеринбург: Изд-во Уральского государственного университета путей сообщения, 2007. – 782 с.

67. Таугер В.М. Основы конструирования мехатронных модулей и систем: учебное пособие / В.М. Таугер. – Екатеринбург: УрГУПС, 2004. – 140 с.
68. Подураев Ю.В. Анализ и проектирование мехатронных систем на основе критерия функционально-структурной интеграции / Ю.В. Подураев // Мехатроника, автоматизация, управление. – 2002. – № 4. – С. 6–11.
69. Бошляков А.А. Проектирование алгоритмического и программного обеспечения мехатронных систем / А.А. Бошляков, С.В. Овсянников. - 2007. – Электронный ресурс http://hoster.bmstu.ru/~sm7/Metodiky/Proektirovanie/Proekt_algoritm_prigr_obesp_mehsyst.doc
70. Электронный ресурс - <http://methodist.lbz.ru/authors/informatika/2/files/pk/kl11gl2.pdf>
71. Электронный ресурс - http://ru.wikipedia.org/wiki/Визуальное_программирование
72. Балабанов В.И. Нанотехнологии. Наука будущего / В.И. Балабанов. – М.: Эксмо, 2009. - 246 с.
73. Кобаяси Н. Введение в нанотехнологию / Н. Кобаяси; пер. с японск. – 2-е изд. – М.: БИНОМ. Лаборатория знаний, 2008. – 134 с.
74. Шахинпур М. Курс робототехники / М. Шахинпур. – пер. с англ. М.: Мир, 1990. – 527 с.
75. Зенкевич С.Л. Управление роботами. Основы управления манипуляционными роботами: учебник для вузов / С.Л. Зенкевич, А.С. Ющенко. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2000. – 400 с.
76. Электронный ресурс - <http://robocraft.ru/blog/mechanics/756.html>

ЗМІСТ

ВСТУП.....	3
1. ПРИСТРОЇ МЕХАТРОННИХ СИСТЕМ.....	12
1.1. Датчики.....	12
1.1.1. Класифікація датчиків.....	12
1.1.2. Основні характеристики датчиків.....	13
1.1.3. Види датчиків та принципи їх роботи.....	14
1.1.3.1. Електроконтактні датчики.....	14
1.1.3.2. Потенціометричні датчики.....	14
1.1.3.3. Тензометричні датчики.....	15
1.1.3.4. П'єзоелектричні датчики.....	16
1.1.3.5. Датчик Холла.....	17
1.1.3.6. Ємнісні датчики.....	18
1.1.3.7. Оптичні датчики.....	19
1.1.3.8. Електромагнітні датчики.....	21
1.1.3.9. Ультразвукові датчики.....	23
1.1.3.10. Датчики температури.....	24
1.1.3.11. Датчики вологості.....	25
1.1.4. Аналогова та цифрова (дискретна) форма представлення даних.....	25
1.1.5. „Інтелектуальні” датчики.....	27
1.2. Привод мехатронних систем.....	30
1.2.1. Вимоги до приводів.....	30
1.2.2. Гідравлічні і пневматичні приводи.....	31
1.2.3. Електропривод. Основні поняття та класифікація.....	31
1.2.4. Електропривод постійного струму.....	33
1.2.4.1. Колекторний електропривод.....	33
1.2.4.2. Кроковий електропривод.....	37
1.2.5. Електропривод змінного струму.....	40
1.2.5.1. Частотне скалярне керування асинхронним двигуном.....	40
1.2.5.2. Частотне векторне керування асинхронним двигуном.....	42
1.2.5.3. Будова та принцип дії частотного перетворювача.....	47

1.2.5.4. Основні функції частотного перетворювача.....	58
1.2.6. Модулі руху в мехатронних система.....	64
1.2.6.1. Електромеханічні модулі руху.....	65
1.2.6.2. Мехатронні модулі руху.....	66
1.2.6.3. Інтелектуальні модулі руху.....	67
1.3. Контролерна та комп'ютерна техніка у мехатронних системах.....	70
1.3.1. Історичний розвиток мікроконтролерів та сфери їх застосування у сучасному виробництві.....	70
1.3.2. Програмовані логічні контролери.....	73
1.3.3. Архітектура мікроконтролера.....	76
1.3.4. Характеристики мікроконтролерів.....	84
1.3.5. Пристрої збору даних.....	88
1.3.6. Комп'ютер у мехатронних системах.....	92
1.3.6.1. Комп'ютер як контролер.....	93
1.3.6.2. Комп'ютер для спілкування з оператором.....	95
1.3.6.3. Промислові комп'ютери.....	96
1.3.7. Пристрої вводу-виводу.....	98
1.4. Питання та завдання для самоперевірки.....	101
2. МЕТОДИ КЕРУВАННЯ МЕХАТРОННИМИ СИСТЕМАМИ.....	102
2.1. Ієрархія систем керування у мехатроніці.....	102
2.2. Проектування систем керування мехатронними об'єктами.....	109
2.3. Основні поняття про будову та алгоритми роботи автоматичних систем керування мехатронними об'єктами.....	113
2.3.1. Основні поняття теорії керування.....	113
2.3.2. Стійкість динамічної системи.....	115
2.3.3. Класифікація завдань автоматичного керування... ..	118
2.3.4. Показники якості автоматичного регулювання....	123
2.4. ПІД-регулятор: теорія, особливості реальних пристроїв та методи їх налаштування.....	124
2.4.1. Історія виникнення ПІД-регулятора.....	124
2.4.2. Класичний ПІД-регулятор та його частинні випадки.....	128
2.4.2.1. Загальна структура системи із регулятором.....	128

2.4.2.2. Закони регулювання.....	129
2.4.3. Особливості реальних регуляторів.....	132
2.4.4. Методи налаштування ПІД-регуляторів.....	137
2.4.4.1. Ручні методи.....	137
2.4.4.2. Метод Зіглера-Нікольса.....	138
2.4.4.3. Автоматичне налаштування та адаптація.....	141
2.4.4.4. Табличне налаштування регулятора.....	144
2.5. Оптимальне керування мехатронними системами.....	145
2.5.1. Вступ. Історичний розвиток екстремальних задач.....	145
2.5.2. Постановка задач оптимального керування та їх класифікація.....	147
2.5.3. Класичне варіаційне числення.....	153
2.5.3.1. Задача із закріпленими кінцями й фіксованим часом.....	153
2.5.3.2. Задача з незакріпленими кінцями й фіксованим часом.....	159
2.5.4. Принцип максимуму Понтрягіна.....	164
2.5.5. Метод динамічного програмування.....	166
2.5.6. Наближені методи синтезу оптимального керування.....	170
2.6. Інтелектуальні мехатронні системи керування.....	171
2.6.1. Основні поняття штучого інтелекту. Аналогії з процесами мислення людини.....	171
2.6.2. Використання систем з нечіткою логікою.....	174
2.6.3. Основи нечіткої логіки (Fuzzy Logic).....	175
2.6.4. Структура й алгоритм фаззі-керування.....	179
2.6.5. Реалізація фаззі-керування.....	181
2.6.6. Особливості фаззі-керування.....	186
2.6.7. Загальна інформація про штучні нейрони.....	188
2.6.8. Архітектура штучних нейронних мереж.....	192
2.6.9. Навчання штучних нейронних мереж.....	195
2.6.10. Нейрочіпи та нейрокомп'ютери.....	199
2.6.11. Особливості використання штучних нейронних мереж у системах керування.....	201
2.7. Питання та завдання для самоперевірки.....	208

3. СПОСОБИ ОРГАНІЗАЦІЇ ВЗАЄМОДІЇ МІЖ ЕЛЕМЕНТАМИ МЕХАТРОННОЇ СИСТЕМИ.....	209
3.1. Основні функції, будова та параметри системи комп'ютерного керування.....	209
3.2. Способи організації обчислювального пристрою системи комп'ютерного керування.....	212
3.3. Варіанти організації інтерфейсної частини системи комп'ютерного керування.....	218
3.4. Передача даних між мехатронними пристроями.....	224
3.4.1. Загальні відомості про промислові мережі.....	225
3.4.2. Провідникові мережі.....	230
3.4.2.1. Інтерфейси RS-485, RS -422 і RS -232.....	230
3.4.2.2. CAN.....	236
3.4.3. Безпроводні мережі.....	238
3.4.3.1. Области використання та переваги безпроводних мереж.....	238
3.4.3.2. Проблеми безпроводних мереж і шляхи їх розв'язання.....	240
3.4.3.3. Bluetooth.....	244
3.4.3.4. Wi-Fi.....	246
3.4.4. Методи перевірки помилок у отриманих даних...	248
3.5. Питання та завдання для самоперевірки.....	252
 4. ПРОЕКТУВАННЯ ТА ЕКСПЛУАТАЦІЯ МЕХАТРОННИХ ПРИСТРОЇВ.....	253
4.1. Проектування мехатронних систем і пристроїв.....	253
4.1.1 Системний підхід до проектування.....	253
4.1.2. Структура процесу проектування.....	257
4.1.3. Стадії проектування.....	259
4.1.4. Концепція проектування мехатронних модулів і систем.....	263
4.1.5. Алгоритм проектування.....	265
4.1.6. Проектування мехатронної системи із мікропроцесорними пристроями.....	267
4.2. Мехатронні модулі - основа проектування сучасних мехатронних об'єктів.....	270
4.3. Розробка програмної частини мехатронного пристрою.....	276
4.3.1. Рівні програмування.....	276
4.3.2. Парадигми програмування.....	280

4.3.3. Методології й технології програмування.....	282
4.3.4. Огляд програмних засобів, використовуваних при проектуванні програмного забезпечення мікроконтролерів.....	285
4.3.4.1. Програмні засоби для розробки програмного забезпечення мікроконтролерів.....	285
4.3.4.2. Програмні засоби для налаштування і контролю програм мікроконтролерів.....	286
4.4. Нанотехнології в мехатроніці.....	287
4.4.1. Історія розвитку нанотехнології в датах.....	288
4.4.2. Нанотехнології.....	291
4.4.3. Нанороботи.....	292
4.5. Роботосистеми.....	294
4.5.1. Класифікація роботів по показниках, що визначають їхню конструкцію.....	294
4.5.2. Способи керування роботами.....	297
4.5.3. Основні функції робота.....	298
4.5.4. Манипуляційні системи роботів.....	301
4.5.5. Сенсорні системи робота (підсистеми тактильного та зорового чуття).....	305
4.5.6. Циклічне програмне керування.....	309
4.5.7. Системи адаптивного керування.....	311
4.5.8. Дві задачі кінематики маніпуляторів.....	316
4.6. Питання та завдання для самоперевірки.....	322
ДОДАТОК 1.....	323
ДОДАТОК 2.....	333
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	346
ЗМІСТ.....	353