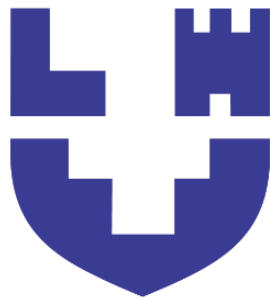


Міністерство освіти і науки України
Луцький національний технічний університет



**АВТОМАТИЗАЦІЯ ТА
КОМП'ЮТЕРНО-ІНТЕГРОВАНІ
ТЕХНОЛОГІЇ**

Магістерський курс

Навчальний посібник

**Гуменюк Л. О., Гуменюк П. О., Пальчевський Б. О.,
Решетило О. М., Сацик В. О., Смолянкін О. О., Федік Л. Ю.**

Загальна редакція к.т.н., доцента Гуменюка П. О.

Луцьк
2024

УДК 004:519.6+681.5:004.89+621.865

А 18

*Рекомендовано до друку вченою радою
Луцького національного технічного університету
та надано гриф «Рекомендовано Луцьким національним технічним університетом»
(протокол № 13 від 29.08.2024 р.)*

Рецензенти:

Ю. М. Кузнєцов – доктор технічних наук, професор, професор кафедри конструювання машин Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського».

В. А. Кирилович – доктор технічних наук, професор кафедри робототехніки, електроенергетики та автоматизації ім. проф. Б. Б. Самотокіна Державного університету «Житомирська політехніка».

Ю. З. Вашкурак – кандидат технічних наук, доцент, доцент кафедри автоматизації та комп'ютерно-інтегрованих технологій Національного університету «Львівська політехніка».

Автоматизація та комп'ютерно-інтегровані технології. Магістерський

курс: навч. посібник / укладачі: Л. О. Гуменюк, П. О. Гуменюк, Б. О. Пальчевський, О. М. Решетило, В. О. Сацик, О. О. Смолянкін, Л. Ю. Федік. Заг. ред. Гуменюка П. О. Луцьк: ЛНТУ, 2024. 205 с.

Навчальний посібник підготовано колективом авторів кафедри автоматизації та комп'ютерно-інтегрованих технологій Луцького національного технічного університету. Посібник висвітлює зміст нормативних освітніх компонент магістерської освітньої програми «Автоматизація та комп'ютерно-інтегровані технології» спеціальності 174 Автоматизація, комп'ютерно-інтегровані технології та робототехніка.

Запитання для перевірки знань та список використаних джерел до кожного розділу сприятимуть поглибленому вивченню матеріалу.

УДК 004:519.6+681.5:004.89+621.865

© Гуменюк Л. О., Гуменюк П. О.,
Пальчевський Б. О., Решетило О. М.,
Сацик В. О., Смолянкін О. О., Федік Л. Ю.
2024

ЗМІСТ

ВСТУП (Гуменюк П. О.)	5
АНАЛІЗ ДАНИХ І СИСТЕМНИЙ ПІДХІД ПРИ РОЗВ'ЯЗАННІ СКЛАДНИХ ЗАДАЧ (укладач Гуменюк Л. О.)	6
ТЕМА 1 МАШИННЕ НАВЧАННЯ	6
ТЕМА 2 КАРТА МАШИННОГО НАВЧАННЯ	8
ТЕМА 3 ДАНІ	13
ТЕМА 4 ПРОЦЕС АНАЛІЗУ ДАНИХ. ПОЧАТКОВІ ЕТАПИ	15
ТЕМА 5 КЛАСИФІКАЦІЯ ТА КЛАСТЕРИЗАЦІЯ	21
ТЕМА 6 МЕТОДИ КЛАСИФІКАЦІЇ І ПРОГНОЗУВАННЯ. ДЕРЕВА РІШЕНЬ	26
ТЕМА 7 МЕТОДИ КЛАСИФІКАЦІЇ І ПРОГНОЗУВАННЯ. МЕТОД «НАЙБЛИЖЧОГО СУСІДА». БАЙЄСОВА КЛАСИФІКАЦІЯ	31
ТЕМА 8 МЕТОДИ КЛАСТЕРНОГО АНАЛІЗУ. АЛГОРИТМ К-СЕРЕДНІХ (K-MEANS)	35
Завдання для перевірки знань.	36
Список використаних джерел.	37
МЕТОДИ ІМІТАЦІЙНОГО МОДЕЛЮВАННЯ ТА ОПТИМІЗАЦІЇ (укладач Гуменюк П. О.)	39
ТЕМА 1 ВИДИ І МЕТОДИ ІМІТАЦІЙНОГО МОДЕЛЮВАННЯ	39
ТЕМА 2 ЕТАПИ РОЗРОБКИ МОДЕЛЕЙ.....	48
ТЕМА 3 ІМІТАЦІЙНЕ МОДЕЛЮВАННЯ СИСТЕМ.....	55
ТЕМА 4 СТАТИСТИЧНЕ МОДЕЛЮВАННЯ. МЕТОД МОНТЕ-КАРЛО.....	61
ТЕМА 5 ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ІМІТАЦІЙНОГО МОДЕЛЮВАННЯ	67
Завдання для перевірки знань.	72
Список використаних джерел.	74
МЕТОДОЛОГІЯ І ПРАКТИКА НАУКОВИХ ДОСЛІДЖЕНЬ ТА ІНТЕЛЕКТУАЛЬНА ВЛАСНІСТЬ (укладач Пальчевський Б. О.)	75
ТЕМА 1 ПОНЯТТЯ ПРО НАУКУ ЯК СИСТЕМУ ЗНАНЬ	75
ТЕМА 2 МЕТОДОЛОГІЧНІ АСПЕКТИ НАУКОВОГО ДОСЛІДЖЕННЯ	80
ТЕМА 3 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ НАУКОВОГО ДОСЛІДЖЕННЯ	85
ТЕМА 4 ВИБІР НАПРЯМКУ ТЕМИ ДОСЛІДЖЕННЯ	87
ТЕМА 5 СТРУКТУРА ПРИКЛАДНОГО НАУКОВОГО ДОСЛІДЖЕННЯ	89
ТЕМА 6 МЕТОДОЛОГІЯ ТЕОРЕТИЧНОГО ДОСЛІДЖЕННЯ	92
ТЕМА 7 ПРОВЕДЕННЯ ЕКСПЕРИМЕНТАЛЬНИХ ДОСЛІДЖЕНЬ	94

ТЕМА 8 СПРОЩЕННЯ ЕКСПЕРИМЕНТУ МЕТОДАМИ ТЕОРІЇ ПОДІБНОСТІ	100
ТЕМА 9 ОФОРМЛЕННЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ	104
Завдання для перевірки знань.	106
Список використаних джерел.	107
РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ УПРАВЛІННЯ ТЕХНОЛОГІЧНИМИ КОМПЛЕКСАМИ, ЗАСОБИ ЛЮДИНО-МАШИННОГО ІНТЕРФЕЙСУ (укладачі Решетило О. М., Смолянкін О. О.).....	109
ТЕМА 1 ОГЛЯД МІКРОПРОЦЕСОРІВ ЇХ КЛАСИФІКАЦІЯ	109
ТЕМА 2 СТРУКТУРА, ПРИНЦИП РОБОТИ І ПРОГРАМУВАННЯ МІКРОКОНТРОЛЕРІВ ADAM-4000.....	112
ТЕМА 3 СТРУКТУРА, ПРИНЦИП РОБОТИ. ПРОГРАМУВАННЯ МІКРОКОНТРОЛЕРІВ ADAM-5000.....	115
ТЕМА 4 СТРУКТУРА, ПРИНЦИП РОБОТИ. ПРОГРАМУВАННЯ МІКРОКОНТРОЛЕРІВ МІК-51	120
Завдання для перевірки знань.	131
Список використаних джерел.	134
ІНТЕЛЕКТУАЛЬНІ МЕТОДИ УПРАВЛІННЯ ТА ЦИФРОВІ ТЕХНОЛОГІЇ (укладач Сацик В. О.)	135
ТЕМА 1 ПОНЯТТЯ ІНТЕЛЕКТУАЛЬНИХ МЕТОДІВ, СИСТЕМ ТА ТЕХНОЛОГІЙ	135
ТЕМА 2 ПІДХОДИ ДО ІНТЕЛЕКТУАЛЬНИХ СИСТЕМ	151
Завдання для перевірки знань.	161
Список використаних джерел.	161
РОБОТОТЕХНІЧНІ ТА ІНТЕЛЕКТУАЛЬНІ МЕХАТРОННІ ПРИСТРОЇ (укладач Федік Л. Ю.).....	162
ТЕМА 1 ХАРАКТЕРИСТИКА РОБОТІВ І МАНІПУЛЯТОРІВ	162
ТЕМА 2 СИСТЕМИ КЕРУВАННЯ ПРОМИСЛОВИХ РОБОТІВ.....	175
ТЕМА 3 МЕХАТРОНІКА І МЕХАТРОННІ СИСТЕМИ.	187
ТЕМА 4 МОДУЛІ РУХУ В МЕХАТРОННИХ СИСТЕМАХ РОБОТІВ.....	197
Завдання для перевірки знань.	204
Список використаних джерел.	205

ВСТУП

(Гуменюк П. О.)

Навчальний посібник охоплює основний зміст навчальних дисциплін з підготовки фахівців за освітнім рівнем «магістр» освітньої програми «Автоматизація та комп'ютерно-інтегровані технології» спеціальності 174 Автоматизація, комп'ютерно-інтегровані технології та робототехніка в Луцькому національному технічному університеті.

В сучасному світі інформаційні технології та автоматизація переплітаються зі всіма аспектами життя, мають потенціал впливати на покращення якості життя людей, вирішувати складні проблеми, поліпшувати процеси та забезпечувати безпеку. Фахівці у цій галузі матимуть можливість створювати нові технології та впливати на майбутнє суспільства.

Розробка автоматизованих систем і роботів вимагає творчого підходу та розв'язання складних технічних завдань. Навчання за спеціальністю 174 Автоматизація, комп'ютерно-інтегровані технології та робототехніка дозволить бути в центрі інноваційного прогресу, працювати з передовими технологіями, такими, як штучний інтелект, машинне навчання, робототехніка та інші.

Метою магістерського курсу є підготовка фахівців, здатних до ефективного виконання професійних завдань та комплексного розв'язання задач і проблем створення, вдосконалення, модернізації, експлуатації та супроводження систем автоматизації у приладобудуванні та електроніці, робототехнічних комплексів, їх компонентів, кіберфізичних систем, технологій цифрової трансформації, що стоять за завданнями Industry 4.0, сприяють процесу швидкої адаптації продукції та послуг підприємств та компаній, а також забезпечують перехід від фізичного світу до цифрового, а також оволодіння знаннями, що необхідні для подальшого навчання та самостійного професійного розвитку.

Навчання на магістерському рівні дозволить здобути компетентності, які допоможуть працювати над інноваційними проектами, творчими викликами і підвищувати професійну майстерність.

Саме для узагальнення теоретичних і практичних положень для майбутніх магістрів з автоматизації, комп'ютерно-інтегрованих технологій та робототехніки і призначено пропонований навчальний посібник. Запитання для перевірки знань та список використаних джерел сприятимуть поглибленому вивченню матеріалу.

Посібник призначений для студентів та викладачів, які цікавляться прийняттям ефективних професійних рішень в області комп'ютерно-інтегрованих технологій, робототехнічних комплексів та розв'язанням актуальних задач і проблем автоматизованих виробництв.

АНАЛІЗ ДАНИХ І СИСТЕМНИЙ ПІДХІД ПРИ РОЗВ'ЯЗАННІ СКЛАДНИХ ЗАДАЧ

(укладач Гуменюк Л. О.)

ТЕМА 1 МАШИННЕ НАВЧАННЯ

Ціль машинного навчання – передбачити результати на основі вхідних даних. Чим більш різноманітними є вхідні дані, тим легше алгоритмам виявити закономірності та точніше зробити прогноз.

Машинне навчання та традиційне програмування

Для кращого розуміння роботи машинного навчання, варто розглянути його відмінності від традиційного програмування.

Машинне навчання не замінює традиційне програмування. Наприклад, дата-інженер не буде використовувати алгоритми машинного навчання для створення веб-сайту. Машинне навчання зазвичай доповнює стандартні програмні інструменти. Головне правило: застосовуйте машинне навчання тоді, коли традиційні методи програмування виявляються неефективними для вирішення певного завдання.

Підхід традиційного програмування

У традиційному програмуванні для отримання рішення інженер розробляє алгоритм і пише код. Після цього він задає вхідні параметри, а алгоритм видає результат (рис. 1.1).



Рисунок 1.1 – Як інженер створює рішення за допомогою традиційного програмування

Таким чином, за допомогою традиційного програмування ми створюємо рішення, яке може обробити набір параметрів. Основна проблема в тому, що

людині дуже складно працювати з великим об'ємом параметрів, тоді як обмежений набір дозволяє побудувати лише дуже просту модель.

Підхід машинного навчання

Для вирішення тієї ж задачі методами машинного навчання дата-інженер використовує зовсім інший підхід. Замість того, щоб розробляти алгоритм самостійно, йому необхідно зібрати масив історичних даних, який буде використаний для напівавтоматичної побудови моделі.

Після збору достатньої кількості даних дата-інженер завантажує їх у різні алгоритми машинного навчання. Результатом є модель, яка може прогнозувати нові результати на основі нових вхідних даних. Весь процес виглядає наступним чином (рис. 1.2).

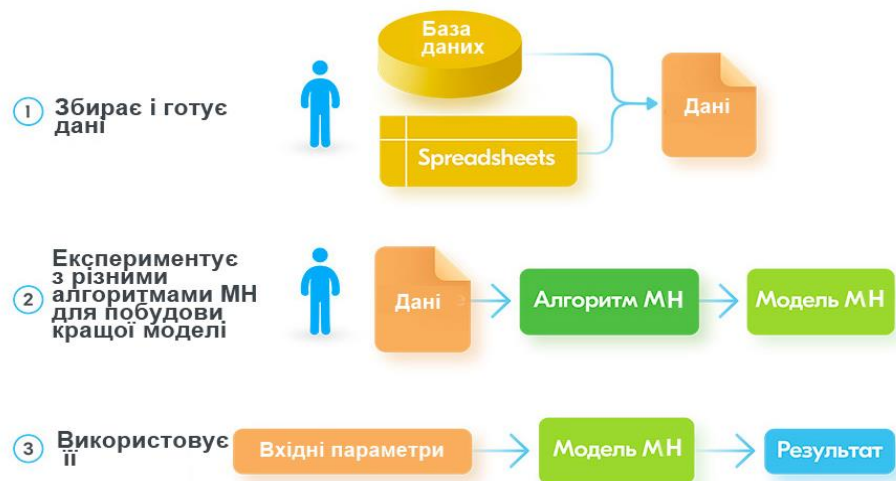


Рисунок 1.2 – Як дата-інженер розробляє рішення за допомогою машинного навчання

Основна відмінність між традиційним програмуванням і машинним навчанням полягає в тому, що в машинному навчанні нам не потрібно будувати модель самостійно – це завдання виконують алгоритми машинного навчання.

Школи машинного навчання

У процесі розвитку науки сформувалося п'ять основних шкіл машинного навчання.

- *Символісти* черпають ідеї з філософії та логіки.
- *Коннекціоністи* надихаються нейробіологією.
- *Еволюціоністи* звертаються до генетики та еволюційної біології.
- *Прихильники байєсівського підходу* вважають, що навчання є різновидом ймовірнісного підходу.

– *Аналогисти* впевнені, що світ складається з аналогій і що наше навчання зводиться до них.

Датасайнтист (від англ. "data scientist") – це вчений, який займається обробкою даних і машинним навчанням.

Таким чином, машинне навчання є наукою про дані та їх обробку, синтезуючи такі науки, як математична статистика, вища математика, паралельні алгоритми, програмна інженерія, теорія ймовірностей, обробка великих даних і математичний аналіз.

ТЕМА 2 КАРТА МАШИННОГО НАВЧАННЯ

В машинному навчанні є всього чотири основні напрями (рис. 2.1).



Рисунок 2.1 – Основні види машинного навчання

2.1 Класичне навчання

Перші алгоритми машинного навчання були розроблені ще в 1950-х роках. Вони вирішували формальні завдання, такі як пошук закономірностей у числах, оцінка близькості точок у просторі та розрахунок напрямків. Незважаючи на свою популярність, класичні алгоритми залишаються досить простими.

Класичне навчання зазвичай поділяють на дві категорії: навчання з вчителем (Supervised Learning) і навчання без вчителя (Unsupervised Learning).

Навчання з вчителем

У цьому типі навчання машина має певного вчителя, який вказує, як правильно. Наприклад, вчитель може показувати картинку з кішкою та

повідомляти, що це кішка, а іншу картинку – з собакою. Вчитель попередньо розділяє всі дані на категорії (наприклад, кішки і собаки), і машина навчається на цих прикладах.

Очевидно, що за наявності вчителя машина навчиться швидше і точніше, тому цей тип навчання використовується значно частіше. Завдання навчання з вчителем поділяються на два типи:

- *класифікація* – передбачення категорії об'єкта;
- *регресія* – передбачення значення на числовій прямій.

Класифікація розділяє об'єкти за відомими ознаками і є найбільш популярним завданням у машинному навчанні. Для класифікації завжди потрібен вчитель – розмічені дані з ознаками і категоріями, які машина буде вчитися розрізняти. Сьогодні для класифікації часто використовують нейронні мережі, оскільки вони були створені саме для цього.

Регресія – це класифікація, але замість категорії передбачається числове значення. Регресія ідеально підходить для завдань, де є залежність від часу. Якщо регресія формує пряму лінію, її називають лінійною, а якщо криву – поліноміальною. Це два основних види регресії, решта методів рідкісні.

Схожість між регресією і класифікацією підтверджується тим, що багато класифікаторів можна перетворити на регресори. Наприклад, замість визначення, до якого класу належить об'єкт, можна вимірювати, наскільки він близький до певного класу – і це буде регресія.

Навчання без вчителя

Навчання без вчителя часто використовується як метод аналізу даних, а не як основний алгоритм.

Кластеризація – це процес класифікації, але без попередньо визначених класів. Машина самостійно знаходить схожі об'єкти та групує їх у кластери. Кількість кластерів може бути задана заздалегідь або визначена машиною. Схожість об'єктів машина визначає за ознаками, які вони мають – ті об'єкти, які мають багато спільних характеристик, об'єднуються в один кластер.

Як і класифікація, кластеризація може використовуватися для виявлення аномалій. Наприклад, якщо поведінка користувача після реєстрації сильно відрізняється від нормальної, його можна заблокувати і перевірити, чи не є він ботом. При цьому не потрібно знати, що є «нормальною поведінкою» – машина сама аналізує всі дії користувачів і визначає, хто є нормальним.

Зменшення розмірності (Узагальнення). Спочатку це були методи, які використовували дата-аналітики, щоб знайти щось цікаве в великих обсягах даних. Коли просто будувати графіки в Excel вже не допомагало, вони розробили методи, що змушують машини шукати закономірності. Так з'явилися методи, які назвали

Dimension Reduction або Feature Learning. Практична користь цих методів полягає в тому, що можна об'єднати кілька ознак в одну і отримати абстракцію.

Цей підхід добре підходить для визначення тематик текстів (Topic Modelling). Можна абстрагуватися від конкретних слів до рівня сенсів навіть без списку категорій. Алгоритм, відомий як Латентно-семантичний аналіз (LSA), базується на ідеї, що частота появи слова в тексті залежить від його тематики: у наукових статтях більше технічних термінів, у новинах про політику – імен політиків. Потрібно об'єднати слова і документи в одну ознаку, щоб не втратити приховані (латентні) зв'язки. Для цього використовується Сингулярне розкладання (SVD), яке легко виявляє корисні тематичні кластери зі слів, що часто зустрічаються разом.

Пошук правил (асоціація) – пошук закономірностей у потоках замовлень.

Серед усіх методів навчання пошук правил є найменш розвиненою категорією. Класичні способи полягають у простому переборі пар всіх куплених товарів за допомогою дерев або множин. Алгоритми можуть шукати закономірності, але не вміють узагальнювати або відтворювати їх на нових прикладах.

2.2 Навчання з підкріпленням

Навчання з підкріпленням застосовують у ситуаціях, де завданням є не аналіз даних, а виживання у реальному середовищі. Це середовище може бути як відеогрою, так і реальним світом, наприклад, автопілот Tesla або роботи-пилососи.

Знання про навколишнє середовище можуть бути корисними для такого робота, але лише як довідкова інформація. Незалежно від обсягу зібраних даних, він не зможе передбачити всі можливі ситуації. Тому його метою є мінімізувати помилки, а не розрахувати всі можливі ходи.

У навчанні з підкріпленням машина не запам'ятовує кожен рух, а намагається узагальнити ситуації, щоб максимально вигідно реагувати на них.

2.3 Ансамблі

Ансамблі та нейромережі є ключовими підходами досягнення прогресу в машинному навчанні, надаючи найточніші результати та використовуються великими компаніями.

Незважаючи на їх високу ефективність, ідея ансамблів є досить простою: якщо об'єднати кілька менш ефективних методів навчання і навчити їх виправляти помилки один одного, то загальна якість такої системи буде значно вищою, ніж якість кожного методу окремо. Найкращі результати досягаються, коли обрані алгоритми є максимально нестабільними та сильно залежать від вхідних даних.

Існує три перевірені способи створення ансамблів.

1. *Стекінг (Stacking)* – це один з найпопулярніших способів використання декількох алгоритмів для вирішення однієї задачі машинного навчання. Навчаються кілька різних алгоритмів, і їх результати передаються на вхід останньому алгоритму, який приймає остаточне рішення, зазвичай за допомогою регресії. Ключове слово тут – «різних» алгоритмів, адже навчати один і той же алгоритм на одних і тих же даних немає сенсу. На практиці стекінг застосовується рідко, оскільки наступні два методи є точнішими.

2. *Беггінг (Bootstrap Aggregating)* – це технологія класифікації, де елементарні класифікатори навчаються та працюють паралельно (незалежно один від одного). Ідея полягає в тому, що класифікатори не виправляють помилки один одного, а компенсують їх при голосуванні. Базові класифікатори повинні бути незалежними, це можуть бути класифікатори, засновані на різних групах методів або ж навчені на незалежних наборах даних.

Здатність паралелізму надає беггінгу перевагу над наступним методом, який працює точніше, але тільки в один потік.

3. *Бустінг (Boosting)* – це процедура послідовної побудови композиції алгоритмів машинного навчання, коли кожен наступний алгоритм прагне компенсувати недоліки всіх попередніх алгоритмів. Алгоритми виконуються послідовно, кожен наступний приділяє особливу увагу тим випадкам, на яких помилився попередній.

Як і в беггінгу, здійснюються вибірки з вихідних даних, але не зовсім випадково. В кожен нову вибірку включається частина тих даних, на яких попередній алгоритм відпрацював неправильно, тобто новий алгоритм довчається на помилках попереднього.

2.4 Нейромережі і глибоке навчання

Нейромережа складається з набору нейронів і зв'язків між ними. Кожен нейрон можна уявити як функцію з багатьма входами та одним виходом. Завдання нейрона – отримати числа на входах, виконати над ними функцію і передати результат на вихід. Зв'язки між нейронами – це канали для передачі сигналів, і кожен зв'язок має свою вагу, що впливає на передачу сигналу.

Нейрони організовані в шари. Нейрони одного шару не пов'язані між собою, але вони з'єднані з нейронами попереднього і наступного шарів. Дані в такій мережі передаються тільки в одному напрямку – від входів першого шару до виходів останнього. Мережа з декількома шарами, де всі нейрони між ними пов'язані, називається перцептроном (MLP) і є найпростішою архітектурою для новачків. Цей метод відомий як BackPropagation або «Метод зворотного поширення похибки».

Глибоке навчання відрізняється від класичних нейромереж новими методами навчання, які дозволяють ефективно працювати з великими мережами. Сьогодні тільки теоретики розрізняють, яке навчання можна вважати глибоким, а яке – ні.

Згорткові Нейромережі (CNN).

Згорткові нейромережі, відомі як CNN, застосовуються для аналізу зображень і відео, розпізнавання облич, перенесення стилю, генерації та покращення якості зображень. Проблема аналізу зображень полягала в складності виділення ознак. Текст можна розділити за реченнями, використовуючи словникові властивості слів. Наприклад, для текстів це легко зробити, але для зображень прийдеться проводити ручну маркування, щоб пояснити машині, де на фото знаходяться вуха чи хвіст котика. Цей процес, відомий як «ручна робота з ознаками», раніше був загальноприйнятим підходом.

Однак згорткові нейромережі дозволяють машинам самостійно вивчати та виділяти такі ознаки, формуючи їх з базових елементів, таких як горизонтальні, вертикальні або діагональні лінії в блоках 8x8 пікселів. Це дозволяє отримувати кілька масивів ліній як прості ознаки об'єктів на зображенні. Подальші шари нейромережі об'єднують ці лінії для створення складніших ознак. Згортка, яка дає назву методу, можна уявити як прошарок нейромережі, де кожен «нейрон» відповідає за визначення певної функції або ознаки об'єкту.

Отже, згорткові нейромережі самостійно визначають характеристики об'єктів, створюють карту ознак і навчаються впізнавати різноманітні об'єкти на зображеннях.

Рекурентні Нейромережі (RNN)

Рекурентні нейромережі, відомі як RNN, використовуються для таких корисних завдань, як машинний переклад текстів і комп'ютерний синтез мови. Вони ефективно вирішують завдання, пов'язані з послідовностями, такими як голосові, текстові або музичні дані.

Текст, мова або музика представлені послідовностями, де кожне слово або звук залежить від попередніх. Викладено навчання мережі проголошувати окремі слова або звуки, використовуючи множину аудіофайлів, розмічених на слова, і вимагаючи від мережі виводити послідовність сигналів, схожих на їхню вимову. Цей процес включає порівняння з оригіналом, наданим диктором, і намагання максимально наблизитися до ідеалу. На початкових етапах такого завдання може використовуватися перцептрон.

Однак перцептрон не має здатності запам'ятовувати свої попередні відповіді, кожен запуск для нього є незалежним. Виникла ідея додати до кожного нейрона пам'ять, що призвело до виникнення рекурентних мереж, де кожен нейрон

запам'ятовує свої попередні стани і використовує їх у наступних ітераціях як додатковий вхід.

ТЕМА 3 ДАНІ

Дані включають факти, тексти, графіки, картинки, звуки, аналогові або цифрові відео-сегменти в широкому розумінні. Вони можуть бути отримані з вимірювань, експериментів або після виконання арифметичних і логічних операцій. Дані потрібно представляти у формі, що дозволяє їх зберігання, передачу і обробку. Іншими словами, дані – це сирі матеріали, які постачальники даних надають і які споживачі використовують для створення інформації на основі даних.

3.1 Набір даних і їх атрибутів

У таблиці 1 представлена *двовимірна* структура, що складається з набору даних.

Двовимірна таблиця «об'єкт-атрибут»					
	Атрибути				
Об'єкти	Код клієнта	Вік	Сімейний стан	Дохід	Клас
	1	18	Single	125	1
	2	22	Married	100	1
	3	30	Single	70	1
	4	32	Married	120	1
	5	24	Divorced	95	2
	6	25	Married	60	1

По вертикалі таблиці розташовані атрибути об'єктів або їх характеристики, тоді як по горизонталі – самі об'єкти.

Об'єкт описується як набір цих атрибутів, відомий також як запис, випадок або рядок таблиці.

Атрибут представляє собою властивість, що характеризує об'єкт, наприклад, колір очей людини чи температура води. Іноді атрибут також називають змінною, полем таблиці або характеристикою.

Змінна (variable) є загальною властивістю, яка може змінюватися від об'єкта до об'єкта в рамках всіх досліджуваних об'єктів.

Значення змінної (value) є конкретним виявом цієї характеристики. Аналіз великих обсягів даних є складним і часом затратним процесом, але необхідним для отримання достовірних висновків.

Генеральна сукупність (population) включає всі досліджувані об'єкти, тоді як вибірка є частиною цієї сукупності, відібраною для дослідження і отримання висновків про її характеристики.

Вибірка (sample) – частина генеральної сукупності, певним способом відібрана з метою дослідження і отримання висновків про властивості та характеристики генеральної сукупності.

Параметри представляють числові характеристики генеральної сукупності, в той час як статистики є числовими характеристиками вибірки.

Часто дослідження ґрунтуються на гіпотезах. Гіпотези перевіряються за допомогою даних.

Гіпотези є припущеннями щодо параметрів сукупності, які перевіряються за допомогою аналізу даних, наприклад, зв'язок між тривалістю життя та якістю харчування.

3.2 Вимірювання

Вимірювання – це процес присвоєння чисел характеристикам досліджуваних об'єктів відповідно до визначених правил. У процесі підготовки даних здійснюється вимірювання характеристик об'єктів, а не самого об'єкта.

Шкала є правилом, за яким об'єктам присвоюються числа.

Змінні можуть бути числовими або символічними.

Числові дані також можуть бути дискретними або безперервними. Дискретні дані представляють собою значення ознаки, які можна підрахувати за допомогою натуральних чисел, в той час як безперервні дані можуть приймати будь-які значення в певному інтервалі і вимагають високої точності вимірювань.

Існує п'ять типів шкал вимірювання: номінальна, порядкова, інтервальна, відносна і дихотомічна.

Номінальна шкала містить лише категорії, без можливості впорядкування або проведення арифметичних операцій.

Порядкова шкала дозволяє упорядковувати об'єкти за їхніми характеристиками, але не визначає точних відмінностей між ними.

Інтервальна шкала дозволяє обчислювати різниці між значеннями, але не має абсолютних нульових значень. Відносна шкала має точку відліку і дозволяє відношення між значеннями.

Дихотомічна шкала містить тільки дві категорії.

Номінальна шкала описує об'єкти за якісними ознаками, порядкові шкали дозволяють упорядковувати спостереження. Інтервальні і відносні шкали дозволяють вимірювати кількісне значення ознаки і проводити арифметичні операції.

3.3 Класифікація видів даних

Класифікація типів даних може бути проведена на основі їхнього походження і призначення.

1. *Реляційні дані*: це дані, що зберігаються в реляційних базах даних у вигляді таблиць, де кожен рядок відповідає конкретному запису, а кожний стовпчик – атрибуту цих записів.

2. *Багатовимірні дані*: це дані, представлені у вигляді кубів OLAP, де вимірювання або вісі є колекціями даних одного типу, що дозволяє структурувати багатовимірну базу даних.

За стабільністю значень під час виконання завдання дані можуть бути класифіковані як:

– *змінні дані*: значення яких змінюються в процесі виконання завдання;

– *постійні дані*: значення яких залишаються незмінними під час виконання завдання, наприклад, математичні константи або координати нерухомих об'єктів;

– *умовно-постійні дані*: значення яких іноді можуть змінюватися, але ці зміни не залежать від процесу вирішення задачі, а визначаються зовнішніми факторами.

За функціями, які вони виконують, дані можуть бути:

– *довідковими*: інформація, що використовується для допомоги прийняття рішень або для підтримки аналітичних процесів;

– *оперативними*: дані, необхідні для поточного функціонування системи або бізнесу;

– *архівними*: дані, які зберігаються для подальшого використання або для виконання правових або регуляторних вимог.

Також важливо розрізняти дані за часовим аспектом:

– *дані за період*: вони характеризують певний період часу, наприклад, денні, щотижневі або щорічні дані;

– *точкові дані*: представляють значення певної змінної в конкретний момент часу.

Крім того, дані можуть бути первинними (зібрані безпосередньо з джерела) або вторинними (похідні від первинних даних, результат обчислень або аналізу).

ТЕМА 4 ПРОЦЕС АНАЛІЗУ ДАНИХ. ПОЧАТКОВІ ЕТАПИ

Процес аналізу даних є свого роду дослідженням. Як будь-яке дослідження, цей процес складається з певних етапів, що включають елементи порівняння, типізації, класифікації, узагальнення, абстрагування, повторення.

Процес аналізу даних нерозривно пов'язаний з процесом прийняття рішень.

Процес аналізу даних включає наступні етапи:

- аналіз предметної області;
- постановка задачі;
- підготовка даних;
- побудова моделей;
- перевірка та оцінка моделей;
- вибір моделі;
- застосування моделі;
- корекція і оновлення моделі.

4.1 Етап 1. Аналіз предметної області

Дослідження – це процес пізнання певної предметної області, об'єкта чи явища з конкретною метою. Він включає спостереження за властивостями об'єктів для виявлення та оцінки важливих закономірних відносин між показниками цих властивостей з точки зору дослідника.

Розв'язання будь-якої задачі у сфері розробки програмного забезпечення має починатися з вивчення предметної області.

Предметна область – це умовно обмежена частина реальності, яка підлягає опису, моделюванню та дослідженню. Вона складається з об'єктів, які розрізняються за властивостями і знаходяться в певних взаємозв'язках або взаємодіють певним чином. Предметна область є частиною реального світу, нескінченною та містить як важливі, так і незначні дані з точки зору проведеного дослідження.

У процесі вивчення предметної області повинна бути створена її модель. Знання з різних джерел необхідно формалізувати за допомогою відповідних засобів. Це можуть бути текстові описи предметної області або спеціалізовані графічні нотації. Існує багато методик опису предметної області, такі як методика структурного аналізу SADT та базована на ній IDEF0, діаграми потоків даних Гейне-Сарсона, методика об'єктно-орієнтованого аналізу UML та інші. Модель предметної області описує процеси, що відбуваються в ній, і дані, що використовуються у цих процесах.

Це перший етап процесу аналізу даних. Успіх подальшої розробки програми аналізу даних залежить від того, наскільки правильно змодельована предметна область.

4.2 Етап 2. Постановка задачі

Постановка задачі аналізу даних включає наступні етапи:

- формулювання задачі;
- формалізація задачі.

Постановка задачі також охоплює опис статичної та динамічної поведінки досліджуваних об'єктів.

Опис статички передбачає характеристику об'єктів та їх властивостей. При описі динаміки описується поведінка об'єктів та причини, які впливають на їхню поведінку.

Технологія аналізу даних не може замінити аналітика і відповісти на питання, які не були поставлені. Тому постановка задачі є важливим етапом процесу аналізу даних, оскільки саме на цьому етапі визначається, яке завдання необхідно вирішити. Іноді етапи аналізу предметної області та постановки задачі об'єднують в один етап.

4.3 Етап 3. Підготовка даних

Підготовка даних є ключовим етапом, від якого залежить якість результатів всього процесу аналізу даних. Важливо пам'ятати, що на підготовку даних може йти до 80% часу, виділеного на проєкт.

Розглянемо цей етап докладніше.

1. Визначення та аналіз вимог до даних.

На цьому етапі здійснюється моделювання даних, тобто визначення та аналіз вимог до них. Вивчаються питання розподілу користувачів (географічного, організаційного, функціонального); питання доступу до даних, необхідних для аналізу; необхідність у зовнішніх і/або внутрішніх джерелах даних; аналітичні характеристики системи (вимірювання даних, основні види вихідних документів, послідовність перетворення інформації тощо).

2. Збір даних.

Наявність в організації сховища даних робить аналіз простішим і ефективнішим, а його використання дешевшим у порівнянні з використанням окремих баз даних. Однак не всі підприємства мають сховища даних. У цьому випадку джерелом вихідних даних є оперативні, довідкові та архівні бази даних, тобто дані з існуючих інформаційних систем. На цьому етапі також здійснюється кодування деяких даних.

При визначенні необхідної кількості даних слід враховувати, чи є дані впорядкованими чи ні.

- Якщо дані впорядковані і ми маємо справу з часовими рядами, бажано знати, чи включають вони сезонну/циклічну компоненту. У разі наявності сезонної/циклової компоненти, необхідно мати дані щонайменше за один сезон/цикл.

- Якщо дані не впорядковані, тобто події з набору даних не пов'язані з часом, слід дотримуватися таких правил:

- кількість записів у наборі;
- співвідношення кількості записів і кількості вхідних змінних.

Набір даних повинен бути репрезентативним, охоплювати якомога більше можливих ситуацій і відповідати реальній ситуації пропорційністю представлення різних прикладів.

3. Попередня обробка даних.

Для забезпечення якісного аналізу необхідно провести попередню обробку даних. Дані, отримані після збору, повинні відповідати певним критеріям якості, тому оцінка якості даних є важливим етапом цього процесу.

Якість даних (Data quality) – це критерій, що визначає повноту, точність, своєчасність і можливість інтерпретації даних. Дані можуть бути високої або низької якості, причому останні часто називають брудними або "поганими" даними.

Дані високої якості – це повні, точні, своєчасні дані, які легко інтерпретувати. Такі дані забезпечують отримання якісних результатів, які можуть підтримувати процес прийняття рішень.

Дані низької якості (брудні дані) – це неповні, неточні або непотрібні дані, які не мають практичної цінності (наприклад, дані у неправильному форматі, що не відповідає стандартам). Брудні дані можуть з'являтися через різні причини, такі як помилки при введенні, використання різних форматів, невідповідність стандартам, відсутність своєчасного оновлення, невдале оновлення копій даних або невдале видалення дублікатів записів.

Найпоширеніші види брудних даних включають:

- пропущені значення;
- дублікати даних;
- шуми і викиди.

Пропущені значення (Missing Values).

Деякі значення можуть бути відсутні з таких причин:

- дані не були зібрані (наприклад, прихований вік при анкетуванні);
- деякі атрибути можуть бути непридатними для певних об'єктів (наприклад, атрибут «річний дохід» для дитини).

Можливі дії з пропущеними даними:

- виключити об'єкти з пропущеними значеннями з обробки;
- розрахувати нові значення для пропущених даних;
- ігнорувати пропущені значення в процесі аналізу;
- замінити пропущені значення на можливі значення.

Дублювання даних (Duplicate Data).

Набір даних може містити продубльовані записи, тобто дублікати, які мають однакові значення всіх атрибутів. Іноді дублікати додаються свідомо для підвищення значущості деяких записів або виділення певних даних. Проте зазвичай дублікати виникають через помилки під час підготовки даних.

Як можна працювати з продубльованими даними?

Є два основні підходи до обробки дублікатів. Перший підхід полягає у видаленні всієї групи записів, що містять дублікати, що використовується, коли наявність дублікатів підриває довіру до даних і знецінює інформацію. Другий підхід полягає в заміні групи дублікатів на один унікальний запис.

Шуми і викиди.

Викиди – це об'єкти або спостереження, які різко відрізняються від решти даних. Вони є поширеною проблемою в аналізі даних. Викиди можуть бути як окремими спостереженнями, так і груповими аномаліями. Завдання аналітика – не лише виявити викиди, але й оцінити ступінь їхнього впливу на результати подальшого аналізу. Якщо викиди містять важливу інформацію, застосовуються робастні методи і процедури.

Поширена практика двоетапного аналізу передбачає виконання аналізу з викидами і без них, а потім порівняння отриманих результатів. Візуалізація даних, включаючи викиди, у графічному вигляді також може бути корисною.

Очевидно, що результати аналізу на основі брудних даних не можуть бути надійними і корисними. Проте наявність брудних даних не обов'язково вимагає їх очищення чи запобігання їхній появі. Важливо зважити наявність брудних даних проти вартості і часу, необхідних для їх очищення.

Очищення даних.

Очищення даних (data cleaning, data cleansing або scrubbing) займається виявленням і видаленням помилок і невідповідностей у даних, щоб покращити їх якість.

Проблеми з якістю даних зустрічаються в окремих наборах даних, таких як файли і бази даних. Коли необхідно інтегрувати безліч джерел даних, як у сховищах, інтегрованих системах баз даних або глобальних інформаційних системах Інтернету, потреба в очищенні даних суттєво зростає. Це відбувається тому, що джерела часто містять розрізнені дані у різному поданні. Для забезпечення доступу до точних і узгоджених даних потрібна консолідація різних подань даних і виключення дубльованої інформації.

Метод очищення даних повинен відповідати ряду критеріїв.

1. Виявляти і видаляти всі основні помилки і невідповідності, як в окремих джерелах даних, так і при інтеграції кількох джерел.

2. Підтримуватися інструментами, щоб зменшити обсяги ручної перевірки та програмування, і бути гнучким для роботи з додатковими джерелами.

3. Очищення даних не повинно проводитися окремо від пов'язаних зі схемою перетворення даних, які виконуються на основі складних метаданих.

4. Інфраструктура технологічного процесу повинна інтенсивно підтримуватися для сховищ даних, забезпечуючи ефективно і надійне виконання всіх етапів перетворення для множини джерел і великих наборів даних.

Етапи очищення даних.

Очищення даних включає наступні етапи.

1. Аналіз даних.
2. Визначення порядку і правил перетворення даних.
3. Підтвердження.
4. Перетворення.
5. Противотік очищених даних.

Етап 1. Аналіз даних.

Докладний аналіз даних необхідний для виявлення типів помилок і невідповідностей. Тут можна використовувати як ручну перевірку даних або їх шаблонів, так і спеціальні програми для отримання метаданих про властивості даних та визначення проблем з якістю.

Етап 2. Визначення порядку і правил перетворення даних.

Залежно від кількості джерел даних, ступеня їх неоднорідності та забрудненості, дані можуть вимагати значного перетворення та очищення. Іноді для узгодження джерел з загальною моделлю даних використовується трансляція схеми; для сховищ даних зазвичай використовується реляційне уявлення. Початкові етапи очищення можуть уточнити або змінити опис проблем окремих джерел даних, а також підготувати дані для інтеграції. Наступні кроки повинні бути спрямовані на інтеграцію схем/даних і усунення проблем множинних елементів, таких як дублікати. Перетворення даних, як і етапи очищення, повинні визначатися за допомогою декларативного запиту, що забезпечує автоматичну генерацію коду перетворення. Крім того, в процесі перетворення повинна бути можливість запуску написаного користувачем коду очищення та спеціальних засобів. Етапи перетворення можуть вимагати зворотного зв'язку з користувачем щодо тих елементів даних, для яких відсутня вбудована логіка очищення.

Етап 3. Підтвердження.

На цьому етапі перевіряється правильність і ефективність процесу перетворення даних. Це здійснюється шляхом тестування та оцінювання на прикладі або на копії даних джерела, щоб виявити можливі покращення. Процес

може вимагати численних ітерацій, оскільки деякі помилки можуть проявитися лише після проведення певних перетворень.

Етап 4. Перетворення.

Цей етап включає виконання визначених перетворень даних.

Етап 5. Протivotік очищених даних.

Після видалення помилок у окремих джерелах, очищені дані повинні замінити забруднені у вихідних джерелах. Це необхідно для того, щоб поліпшені дані потрапили в успадковані додатки і надалі не вимагали додаткового очищення. Для сховищ дані зберігаються в області зберігання даних.

Процес перетворення потребує значних обсягів метаданих (схем, характеристик даних, визначень процесу тощо). Для узгодженості, гнучкості та простоти використання в майбутньому, ці метадані повинні зберігатися в репозиторії на основі СУБД. Для забезпечення якості даних, детальна інформація про процес перетворення повинна записуватися як у репозиторій, так і в трансформовані елементи даних. Це включає інформацію про повноту та актуальність вихідних даних, походження інформації про джерела та внесені зміни.

ТЕМА 5 КЛАСИФІКАЦІЯ ТА КЛАСТЕРИЗАЦІЯ

5.1 Класифікація. Завдання класифікації

Класифікація – це систематичне розподілення предметів, явищ, або процесів за родами, видами, або типами на основі певної істотної ознаки для полегшення їх дослідження. Це угруповання вихідних понять і розташування їх у певному порядку, що відображає ступінь їх схожості.

Класифікація є впорядкованим набором об'єктів, які мають подібні класифікаційні ознаки (одну або кілька властивостей), обраних для визначення подібності або відмінності між цими об'єктами.

При проведенні класифікації необхідно дотримуватися наступних правил.

- Використовувати лише одне обґрунтування для кожного поділу.
- Розподіл має бути співрозмірним, тобто загальний обсяг видових понять повинен дорівнювати обсягу діленого виду.
- Члени поділу повинні взаємно виключати один одного, тобто їх обсяги не повинні перетинатися.
- Розподіл має бути послідовним.

Класифікації бувають:

- *допоміжні (штучні)* – виконуються за зовнішньою ознакою і слугують для впорядкування безлічі предметів (процесів, явищ);

- *природні* – здійснюються за істотними ознаками, що характеризують внутрішню спільність предметів і явищ. Вони є результатом і важливим засобом наукового дослідження, оскільки передбачають і закріплюють результати вивчення закономірностей об'єктів, що класифікуються.

Види класифікації залежать від обраних ознак і процедур поділу.

- Прості класифікації – розподіл родового поняття за ознакою лише один раз до повного розкриття всіх видів, наприклад, дихотомія, де членами поділу є два поняття, що суперечать одне одному («А і не А»).

- Складні класифікації – використовуються для поділу одного поняття за різними основами і синтезу таких розподілів в єдине ціле, наприклад, періодична система хімічних елементів.

Класифікація – це процес віднесення об'єктів (спостережень, подій) до одного з заздалегідь відомих класів. Вона є закономірністю, яка дозволяє робити висновки про визначення характеристик конкретної групи. Для цього необхідні ознаки, що характеризують групу, до якої належить подія або об'єкт, і на основі аналізу вже класифікованих подій формулюються певні правила.

Класифікація відноситься до стратегії навчання з учителем (supervised learning), яку також називають контрольованим або керованим навчанням. Завдання класифікації часто полягає у передбаченні категоріальної залежної змінної (категорії) на основі вибірки безперервних і/або категоріальних змінних.

Інший тип класифікації виникає, коли залежна змінна може приймати значення з деякого набору визначених класів. У цих випадках розглядається множина класів для залежної змінної.

Класифікація може бути одновимірною (за однією ознакою) і багатовимірною (за двома або більше ознаками).

5.2 Процес класифікації

Метою процесу класифікації є створення моделі, яка використовує прогнозуючі атрибути як вхідні параметри для визначення значень залежного атрибута. Процес класифікації полягає в поділі множини об'єктів на класи за певним критерієм.

Класифікатором називається сутність, яка визначає, до якого з визначених класів належить об'єкт на основі вектора ознак.

Для виконання класифікації за допомогою математичних методів необхідно мати формальний опис об'єкта, яким можна оперувати, використовуючи математичний апарат класифікації. Таким описом у нашому випадку є база даних. Кожен об'єкт (запис бази даних) містить інформацію про певні властивості об'єкта.

Набір вихідних даних (або вибірка даних) ділиться на два множини: навчальну та тестову.

– Навчальна множина (training set) включає дані, що використовуються для навчання (побудови) моделі. Ця множина містить вхідні та вихідні (цільові) значення прикладів, призначені для навчання моделі.

– Тестова множина (test set) також містить вхідні та вихідні значення прикладів, але вихідні значення тут використовуються для перевірки працездатності моделі.

Процес класифікації складається з двох етапів: побудови моделі та її використання.

1. Побудова моделі.

– Опис множини визначених класів.
– Кожен приклад набору даних належить до одного визначеного класу.
– На цьому етапі використовується навчальна множина для побудови моделі.

– Отримана модель представлена класифікаційними правилами, деревом рішень або математичною формулою.

2. Використання моделі.

– Класифікація нових або невідомих значень.
– Оцінка правильності (точності) моделі.

Якщо точність моделі допустима, її можна використовувати для класифікації нових прикладів, клас яких невідомий.

5.3 Методи, які застосовуються для вирішення задач класифікації

Для класифікації використовуються різні методи. Основні з них:

- класифікація за допомогою дерев рішень;
- байєсівська (наївна) класифікація;
- класифікація за допомогою штучних нейронних мереж;
- класифікація методом опорних векторів;
- статистичні методи, зокрема, лінійна регресія;
- класифікація за допомогою методу найближчого сусіда;
- класифікація СBR-методом;
- класифікація за допомогою генетичних алгоритмів.

5.4 Точність класифікації: оцінка рівня помилок

Оцінка точності класифікації може бути проведена за допомогою крос-перевірки.

Крос-перевірка (Cross-validation) є процедурою, яка оцінює точність класифікації на тестовій множині даних, що також відома як перевірна множина. Точність класифікації на тестовій множині порівнюється з точністю класифікації на навчальній множині. Якщо результати класифікації на тестовій множині приблизно однакові за точністю з результатами на навчальній множині, то модель вважається успішною після крос-перевірки.

Ділення на навчальну і тестову множини здійснюється шляхом розділення вибірки у певному співвідношенні, наприклад, дві третини даних у навчальну множину та одна третина у тестову. Цей метод рекомендується для вибірок із великою кількістю прикладів. У випадку невеликих вибірок рекомендується використовувати спеціальні методи, де навчальна і тестова множини можуть частково перетинатися.

5.5 Оцінювання класифікаційних методів

Оцінка методів повинна здійснюватися з урахуванням таких характеристик: швидкість, робастність, інтерпретованість та надійність.

Швидкість визначає час, необхідний для створення моделі та її використання.

Робастність означає стійкість методу до будь-яких відхилень вихідних передумов, зокрема, здатність працювати з шумними даними і випадково відсутніми значеннями.

Інтерпретованість забезпечує можливість аналізу та розуміння моделі аналітиком.

Надійність методів класифікації передбачає їх здатність працювати з даними, що містять шуми і викиди.

5.6 Кластеризація. Завдання кластеризації

Кластеризація призначена для групування об'єктів у сукупності на однорідні групи, які називаються кластерами або класами. Якщо розглядати дані вибірки як точки у просторі ознак, то основна мета кластеризації полягає у визначенні «згущень точок».

Кластеризація є описовою процедурою, яка не робить статистичних висновків, але дозволяє провести розвідувальний аналіз і дослідити «структуру даних». Кластер можна розглядати як групу об'єктів, які мають спільні властивості. Основними характеристиками кластера є внутрішня однорідність та зовнішня ізольованість.

Кластери можуть бути ексклюзивними (які не перекриваються, non-overlapping, exclusive) або такими, що перекриваються (overlapping).

Різні методи кластерного аналізу можуть працювати з різними типами даних і мати різну чутливість до шумів або викидів. Вибір методу кластеризації повинен залежати від конкретних особливостей даних і постановки завдання.

На сьогоднішній день існує багато різних алгоритмів кластеризації. Ось деякі з них.

– Алгоритми, засновані на поділі даних (Partitioning algorithms): ці алгоритми включають ітеративні методи, які спочатку розбивають об'єкти на кілька кластерів, а потім вдосконалюють розподіл для покращення кластеризації.

– Ієрархічні алгоритми (Hierarchy algorithms): такі методи використовуються для створення ієрархічної структури кластерів, де кожен об'єкт спочатку є окремим кластером, які потім об'єднуються в більші кластери на основі їх взаємного подібності.

– Методи, засновані на концентрації об'єктів (Density-based methods): ці методи використовують можливість об'єднання об'єктів, ігноруючи шуми і знаходячи кластери різних форм.

– Грід-методи (Grid-based methods): такі методи квантують об'єкти у грід-структури, що дозволяє ефективно визначати кластери.

– Модельні методи (Model-based): ці методи використовують моделі для знаходження кластерів, які найкраще відповідають даним.

Кожен з цих підходів має свої переваги та обмеження, тому вибір відповідного методу кластеризації повинен базуватися на конкретних вимогах задачі та особливостях даних.

Оцінка якості кластеризації може бути проведена за допомогою наступних процедур.

1. Ручна перевірка: аналітик оцінює сформовані кластери на основі своїх знань і досвіду.

2. Встановлення контрольних точок і перевірка на отриманих кластерах: визначаються контрольні точки, для яких відомі правильні класи або кластери, і перевіряється, чи відповідають їм отримані результати.

3. Визначення стабільності кластеризації шляхом додавання нових змінних: досліджується, як змінюється структура кластерів при включенні додаткових ознак у модель.

4. Створення і порівняння кластерів з використанням різних методів: застосовуються різні методи кластеризації для порівняння отриманих результатів. Схожість сформованих кластерів при різних методах свідчить про правильність кластеризації.

Процес кластеризації.

Процес кластеризації є ітеративним і залежить від обраного методу. Отримані результати потребують подальшого аналізу, інтерпретації і дослідження характеристик об'єктів для точного опису сформованих кластерів.

Застосування кластерного аналізу.

Кластерний аналіз застосовується у різних галузях. Наприклад, в медицині для класифікації захворювань, лікування симптомів і таксономії пацієнтів; в археології для класифікації артефактів; в маркетингу для сегментації конкурентів і споживачів; у менеджменті для розподілу персоналу та класифікації постачальників; у соціології для розбиття респондентів на групи з схожими характеристиками.

ТЕМА 6 МЕТОДИ КЛАСИФІКАЦІЇ І ПРОГНОЗУВАННЯ. ДЕРЕВА РІШЕНЬ

Метод дерев рішень (decision trees) є одним з найпопулярніших інструментів для вирішення завдань класифікації та прогнозування. Іноді його також називають деревами рішучих правил або деревами класифікації і регресії.

Цей метод застосовується для класифікації, коли залежна змінна приймає дискретні значення, або для чисельного прогнозування, коли залежна змінна є безперервною.

У найпростішому випадку дерево рішень представляє собою ієрархічну структуру, де кожен вузол має питання з відповідями «Так» або «Ні».

Проходження від кореня до листа дерева дозволяє здійснювати класифікацію, вибираючи один з класів. Кожен внутрішній вузол відповідає атрибуту даних, що називається прогнозуючим атрибутом або атрибутом розщеплення. Листя дерева містять мітки класів, які є значеннями залежної категоріальної змінної.

Дерева можуть бути бінарними, де кожна гілка виходить з вузла в два напрямки, або мають більш складну структуру з більшим числом гілок. Якість побудованого дерева значно залежить від правильного вибору критерію розщеплення, який визначається унікальним шляхом до кожного листа.

Метод дерев рішень, не дивлячись на свою «наївність», є одним із найефективніших для вирішення задач класифікації, завдяки чому він широко застосовується в різних галузях науки та бізнесу.

6.1 Переваги дерев рішень

Класифікаційні моделі у формі дерев рішень відзначаються своєю інтуїтивною зрозумілістю, що спрощує розуміння проблеми, яку вони вирішують. Наприклад, порівняно з «чорними ящиками» такими, як нейронні мережі, дерева рішень легко інтерпретуються користувачем. Ця особливість не лише важлива при класифікації нових об'єктів, але й корисна для розуміння всієї моделі класифікації в цілому.

Дерева рішень дозволяють зрозуміти та пояснити, чому конкретний об'єкт належить до певного класу.

Ці моделі також дають змогу отримувати правила з бази даних у природній мові. Вони особливо корисні в галузях, де складно формалізувати аналітику знань. Наприклад, алгоритм побудови дерев рішень не вимагає від користувача вибору вхідних атрибутів. Замість цього, на вхід можуть подаватись всі існуючі атрибути, і алгоритм сам вибере найбільш значущі для побудови дерева. Це відрізняє його від нейронних мереж, де вибір кількості вхідних атрибутів впливає на час навчання.

Більшість алгоритмів побудови дерев рішень також мають можливість обробки пропущених значень, що підвищує їхню робастність. Вони працюють як з числовими, так і з категоріальними типами даних, що робить їх універсальними у порівнянні з багатьма класичними статистичними методами, які часто обмежені роботою лише з числовими даними.

Дерева рішень будують непараметричні моделі, що відрізняє їх від більшості параметричних статистичних методів. Це означає, що вони можуть успішно застосовуватись у завданнях аналізу даних, де відсутня апріорна інформація про вид залежності між змінними.

6.2 Процес конструювання дерева рішень

Задача класифікації, яка описана, є прикладом стратегії навчання з учителем. У таких випадках всі об'єкти у навчальному наборі даних заздалегідь віднесені до одного з визначених класів. Алгоритми побудови дерев рішень складаються з етапів «побудова» або «створення» дерева (tree building) і «скорочення» дерева (tree pruning). Під час побудови дерева вирішуються питання вибору критерію розщеплення і зупинки навчання, якщо це передбачено алгоритмом. На етапі скорочення дерева розглядається питання відсікання деяких його гілок.

Критерій розщеплення.

Задача класифікації, яка описана, є прикладом стратегії навчання з учителем. У таких випадках всі об'єкти у навчальному наборі даних заздалегідь віднесені до одного з визначених класів. Алгоритми побудови дерев рішень складаються з етапів «побудова» або «створення» дерева (tree building) і «скорочення» дерева

(tree pruning). Під час побудови дерева вирішуються питання вибору критерію розщеплення і зупинки навчання, якщо це передбачено алгоритмом. На етапі скорочення дерева розглядається питання відсікання деяких його гілок.

Процес створення дерева відбувається зверху вниз, тобто є низхідним. Під час цього процесу алгоритм повинен знайти такий критерій розщеплення, щоб розбити множину на підмножини, кожна з яких асоціюється з певним вузлом перевірки. Кожен вузол перевірки пов'язаний з певним атрибутом, який вибирається на підставі того, як ефективно він розбиває вихідну множину даних на підмножини, що представляють один клас або максимально наближені до такого розбиття. Цей підхід спрямований на зниження числа об'єктів інших класів у кожній підмножині, що називається «домішками».

Існують різні критерії розщеплення, серед яких найбільш відомі – міра ентропії і індекс Gini.

В деяких методах використовується міра інформативності підпросторів атрибутів, що базується на ентропійному підході і відома як «міра інформаційного виграшу» або міра ентропії.

Інший популярний критерій – індекс Gini, за яким атрибути вибираються на основі відстаней між розподілами класів.

Велике дерево не означає, що воно «підходяще».

Чим більше окремих випадків описано в дереві рішень, тим менша кількість об'єктів потрапляє в кожен окремий випадок. Такі дерева часто називають «гіллястими» або «кущистими», оскільки вони складаються з великої кількості вузлів і гілок, які розбивають вихідну множину на численні підмножини з дуже обмеженим числом об'єктів. Внаслідок цього «переповнення» таких дерев їх здатність до узагальнення значно зменшується, і побудовані моделі можуть давати неточні результати.

Під час конструювання дерева застосовуються спеціальні процедури для запобігання надмірно великим розмірам. Такі дерева часто називають «деревами відповідних розмірів». Їх мета полягає у тому, щоб дерево було достатньо складним для врахування інформації з досліджуваного набору даних, але водночас достатньо простим для узагальнення результатів. Це означає, що дерево повинно використовувати інформацію, яка поліпшує якість моделі, і ігнорувати ту, що не приносить користі.

Існують дві основні стратегії: перша – це нарощування дерева до певного розміру на основі заданих користувачем параметрів, які можуть ґрунтуватися на досвіді, інтуїції аналітика або деяких «діагностичних повідомленнях» системи.

Друга стратегія полягає у використанні процедур, які автоматично визначають «відповідний розмір» дерева.

Для запобігання створенню надмірно великих дерев використовуються різні процедури, включаючи скорочення дерева шляхом відсікання гілок та використання правил зупинки навчання.

Варто відзначити, що не всі алгоритми конструювання дерева працюють за однаковою схемою: деякі чергують процеси побудови і скорочення дерева, щоб запобігти надмірному зростанню внутрішніх вузлів.

Зупинка побудови дерева.

Зупинка в контексті побудови дерева рішень визначає момент, коли треба завершити подальше розгалуження.

Один з підходів до зупинки, відомий як «рання зупинка» (prepruning), оцінює доцільність подальшого розгалуження на кожному кроці. Використання цього підходу дозволяє зменшити час навчання моделі, але при цьому існує ризик зниження точності класифікації. Тому частіше рекомендується використовувати «відсікання» гілок замість ранньої зупинки.

Інший спосіб зупинки полягає в обмеженні глибини дерева: процес побудови завершується, якщо досягнута задана глибина.

Також можливе визначення мінімальної кількості прикладів, які мають міститися в кінцевих вузлах дерева. У цьому випадку розгалуження продовжуються, поки всі кінцеві вузли не стануть чистими або не будуть містити не більше заданої кількості об'єктів.

Скорочення дерева або відсікання гілок.

Щоб вирішити проблему занадто гіллястого дерева, застосовують процедуру скорочення або відсікання (pruning) гілок. Це означає видалення деяких гілок дерева для поліпшення його узагальнюючої здатності. Після такого скорочення отримане дерево називається усіченим. Якщо усічене дерево ще не є достатньо інтуїтивно зрозумілим, його можна описати набором правил, кожне з яких представляє собою шлях від кореня до листа дерева.

Якість класифікаційної моделі, побудованої за допомогою дерева рішень, характеризується двома основними ознаками: точністю розпізнавання і помилкою.

Точність розпізнавання розраховується як відношення об'єктів, правильно класифікованих в процесі навчання, до загальної кількості об'єктів набору даних, які брали участь в навчанні.

Помилка розраховується як відношення об'єктів, неправильно класифікованих в процесі навчання, до загальної кількості об'єктів набору даних, які брали участь в навчанні.

Відсікання гілок або заміну деяких гілок піддерева слід проводити там, де ця процедура не призводить до зростання помилки. Процес проходить від низу до верху, тобто є висхідним. Дереву, одержувані після відсікання деяких гілок,

називають усіченими. Якщо таке усічене дерево все ще не є інтуїтивним і складним для розуміння, використовують набір правил, які об'єднують в набори для опису класів. Кожен шлях від кореня дерева до його вершини або листа дає одне правило. Умовами правила є перевірки на внутрішніх вузлах дерева.

6.3 Алгоритми, що реалізують дерева рішень

На сьогоднішній день існує велика кількість алгоритмів, що реалізують дерева рішень: CART, C4.5, CHAID, CN2, NewId, ITrule і інші.

Алгоритм CART.

Алгоритм CART (Classification and Regression Tree), як видно з назви, вирішує завдання класифікації і регресії.

Атрибути набору даних можуть мати як дискретне, так і числове значення. Алгоритм CART призначений для побудови бінарного дерева рішень. Бінарні дерева також називають двійковими.

Інші особливості алгоритму CART:

- функція оцінки якості розбиття;
- механізм відсікання дерева;
- алгоритм обробки пропущених значень;
- побудова дерев регресії.

Кожний вузол бінарного дерева при розбитті має тільки двох нащадків, які називаються дочірніми гілками. Подальший поділ гілки залежить від того, чи багато вихідних даних описує ця галузь. На кожному кроці побудови дерева правило, формується в вузлі, ділить задану множину прикладів на дві частини. Права його частина (гілка right) – це та частина множини, в якій правило виконується; ліва (гілка left) – та, для якої правило не виконується.

Функція оцінки якості розбиття, яка використовується для вибору оптимального правила – індекс Gini, описаний вище. Відзначимо, що дана оціночна функція заснована на ідеї зменшення невизначеності в вузлі.

Алгоритми побудови дерев рішень відрізняються такими характеристиками:

- вид розщеплення – бінарне (binary), множинне (multi-way);
- критерії розщеплення – ентропія, Gini, інші;
- можливість обробки пропущених значень;
- процедура скорочення гілок або відсікання;
- можливості вилучення правил з дерев.

ТЕМА 7 МЕТОДИ КЛАСИФІКАЦІЇ І ПРОГНОЗУВАННЯ. МЕТОД «НАЙБЛИЖЧОГО СУСИДА». БАЙЄСОВА КЛАСИФІКАЦІЯ

7.1 Метод «найближчого сусіда» або системи міркувань на основі аналогічних випадків

Метод «найближчого сусіда» (“nearest neighbour”) відноситься до класу методів, робота яких ґрунтується на зберіганні даних в пам'яті для порівняння з новими елементами. При появі нового запису для прогнозування знаходяться відхилення між цим записом і подібними наборами даних, і найбільш подібний (або ближній сусід) ідентифікується.

При такому підході використовується термін «k-найближчий сусід» (“k-nearest neighbour”). Він означає, що вибирається k «верхніх» (найближчих) сусідів для їх розгляду в якості безлічі «найближчих сусідів». Оскільки не завжди зручно зберігати всі дані, іноді зберігається тільки множини «типових» випадків. В такому випадку використовується метод, який називають міркуванням за аналогією (Case Based Reasoning, CBR), міркуванням на основі аналогічних випадків, міркуванням по прецедентах.

Прецедент – це опис ситуації в поєднанні з детальним зазначенням дій, що застосовуються в даній ситуації.

Підхід, заснований на прецедентах, умовно можна поділити на такі етапи:

- збір докладної інформації про поставлену задачу;
- зіставлення цієї інформації з деталями прецедентів, що зберігаються в базі, для виявлення аналогічних випадків;
- вибір прецеденту, найбільш близького до поточної проблеми, з бази прецедентів;
- адаптація обраного рішення до поточної проблеми, якщо це необхідно;
- перевірка коректності кожного отриманого рішення;
- занесення детальної інформації про новий прецедент в базу прецедентів.

Таким чином, висновок, заснований на прецедентах, являє собою такий метод аналізу даних, який робить висновки щодо даної ситуації за результатами пошуку аналогій, що зберігаються в базі прецедентів.

Даний метод по своїй суті відноситься до категорії «навчання без вчителя», завдяки чому робочі характеристики кожної бази прецедентів з плином часу і накопиченням прикладів поліпшуються. Розробка баз прецедентів по конкретній предметній області відбувається на природній для людини мові, отже, може бути виконана найбільш досвідченими співробітниками компанії – експертами або аналітиками, які працюють в даній галузі.

Переваги методу.

- Простота використання отриманих результатів.
- Рішення не унікальні для конкретної ситуації, можливо їх використання для інших випадків.

- Метою пошуку є не гарантовано вірне рішення, а найкраще з можливих.
Недоліки методу «найближчого сусіда».

- Даний метод не створює будь-яких моделей або правил, узагальнюють попередній досвід – у виборі рішення вони ґрунтуються на всьому масиві доступних даних, тому неможливо сказати, на якій підставі будуються відповіді.

- Існує складність вибору «близькості» (метрики). Від цього головним чином залежить обсяг множини записів, які потрібно зберігати в пам'яті для досягнення задовільної класифікації або прогнозу. Також існує висока залежність результатів класифікації від обраної метрики.

- При використанні методу виникає необхідність повного перебору навчальної вибірки при розпізнаванні, наслідок цього – обчислювальна трудомісткість.

- Типові завдання даного методу – це завдання невеликої розмірності за кількістю класів і змінних.

За допомогою даного методу вирішуються завдання класифікації і регресії.

Незалежні і залежні змінні набору даних можуть бути як безперервними, так і категоріальними. Для безперервних залежних змінних завдання розглядається як задача прогнозування, для дискретних змінних – як завдання класифікації.

Передбачення в задачі прогнозування виходить усередненням виходів k -найближчих сусідів, а рішення задачі класифікації засноване на принципі «за більшістю голосів».

Критичним моментом у використанні методу k -найближчих сусідів є вибір параметра k . Він один з найбільш важливих факторів, що визначають якість прогнозу або класифікаційної моделі.

Якщо вибрано дуже маленьке значення параметра k , виникає ймовірність великого розкиду значень прогнозу. Якщо вибране значення занадто велике, це може привести до сильної зміщення моделі. Таким чином, ми бачимо, що повинно бути вибрано оптимальне значення параметра k . Тобто це значення має бути настільки великим, щоб звести до мінімуму ймовірність невірної класифікації, і одночасно, досить малим, щоб k сусідів були розташовані досить близько до точки запиту.

Таким чином, ми розглядаємо k як параметр, для якого повинен бути знайдений компроміс між силою розмаху (розкиду) моделі і її зміщенням.

Оцінка параметра k методом крос-перевірки.

Один з варіантів оцінки параметра k – проведення крос-перевірки.

Крос-перевірка – метод отримання оцінок невідомих параметрів моделі.

Основна ідея методу – поділ вибірки даних на v «складки». В «складки» тут випадковим чином виділені ізольовані підвибірки.

За фіксованим значенням k будується модель k -найближчих сусідів для отримання прогнозів на v -му сегменті (інші сегменти при цьому використовуються як приклади) і оцінюється помилка класифікації. Для регресійних задач найбільш часто в якості оцінки помилки виступає сума квадратів, а для класифікаційних завдань зручніше розглядати точність (відсоток коректно класифікованих спостережень).

Далі процес послідовно повторюється для всіх можливих варіантів вибору v . Після вичерпання v «складок» (циклів), обчислені помилки усереднюються і використовуються в якості міри стійкості моделі (тобто міри якості передбачення в точках запиту). Вищеописані дії повторюються для різних k , і значення, яке відповідає найменшій помилці (або найбільшій класифікаційній точності), приймається як оптимальне (оптимальне в сенсі методу крос-перевірки).

Слід враховувати, що крос-перевірка – обчислювальна процедура, і необхідно надати час для роботи алгоритму, особливо якщо обсяг вибірки досить великий.

Другий варіант вибору значення параметра k – самостійно задати його значення. Однак цей спосіб слід використовувати, якщо є обґрунтовані припущення щодо можливого значення параметра, наприклад, попередні дослідження подібних наборів даних.

Метод k -найближчих сусідів показує досить непогані результати в найрізноманітніших задачах.

7.2 Байєсова класифікація

Спочатку байєсівську класифікація використовувалася для формалізації знань експертів в експертних системах, зараз байєсівську класифікація також застосовується в якості одного з методів аналізу даних.

Так звана наївна класифікація або наївно-байєсовський підхід (naive-bayes approach) є найбільш простим варіантом методу, що використовує байєсовські мережі. При цьому підході вирішуються завдання класифікації, результатом роботи методу є так звані «прозорі» моделі.

«Наївна» класифікація – досить прозорий і зрозумілий метод класифікації. «Наївною» вона називається тому, що виходить з припущення про взаємну незалежності ознак.

Властивості наївною класифікації:

1. Використання всіх змінних і визначення всіх залежностей між ними.

2. Наявність двох припущень щодо змінних:

- всі змінні є однаково важливими;

- всі змінні є статистично незалежними, тобто значення однієї змінної нічого не говорить про значення іншої.

Більшість інших методів класифікації припускають, що перед початком класифікації ймовірність того, що об'єкт належить тому чи іншому класу, однакова, але це не завжди вірно.

Припустимо, відомо, що певний відсоток даних належить до конкретного класу. Виникає питання, чи можемо ми використовувати цю інформацію при побудові моделі класифікації?

Переваги методу:

- в моделі визначаються залежності між усіма змінними, це дозволяє легко обробляти ситуації, в яких значення деяких змінних невідомі;
- байєсовські мережі досить просто інтерпретуються і дозволяють на етапі прогностичного моделювання легко проводити аналіз за сценарієм «що, якщо»;
- байєсовський метод дозволяє природним чином поєднувати закономірності, виведені з даних, і, наприклад, експертні знання, отримані в явному вигляді;
- використання байєсовських мереж дозволяє уникнути проблеми переучування (*overfitting*), тобто надмірного ускладнення моделі, що є слабкою стороною багатьох методів (наприклад, дерев рішень і нейронних мереж).

Недоліки наївно-байєсовського підходу:

- перемножувати умовні ймовірності коректно тільки тоді, коли всі вхідні змінні дійсно статистично незалежні; хоча часто даний метод показує досить хороші результати при недотриманні умови статистичної незалежності, але теоретично така ситуація повинна оброблятися більш складними методами, заснованими на навчанні байєсовських мережах;
- неможлива безпосередня обробка безперервних змінних – потрібно їх перетворення до інтервальної шкали, щоб атрибути були дискретними; однак такі перетворення іноді можуть призводити до втрати значущих закономірностей;
- на результат класифікації в наївно-Байєсовському підході впливають тільки індивідуальні значення вхідних змінних, комбінований вплив пар або трійок значень різних атрибутів тут не враховується. Це могло б поліпшити якість класифікаційної моделі з точки зору її прогностичної точності, однак, збільшило б кількість перевічених варіантів.

ТЕМА 8 МЕТОДИ КЛАСТЕРНОГО АНАЛІЗУ. АЛГОРИТМ К-СЕРЕДНІХ (K-MEANS)

При великій кількості спостережень ієрархічні методи кластерного аналізу не придатні. У таких випадках використовують неієрархічні методи, засновані на поділі, які представляють собою ітеративні методи дроблення вихідної множини. У процесі поділу нові кластери формуються до тих пір, поки не буде виконано правило зупинки.

Така неієрархічна кластеризація полягає в поділі набору даних на певну кількість окремих кластерів. Існує два підходи. Перший полягає у визначенні меж кластерів, як найбільш щільних ділянок в багатовимірному просторі вихідних даних, тобто визначення кластера там, де є велике «згущення точок». Другий підхід полягає в мінімізації міри відмінності об'єктів.

Алгоритм k-середніх (k-means).

Найбільш поширений серед неієрархічних методів алгоритм k-середніх, також званий швидким кластерним аналізом. На відміну від ієрархічних методів, які не вимагають попередніх припущень щодо числа кластерів, для можливості використання цього методу необхідно мати гіпотезу про найбільш ймовірне кількість кластерів.

Алгоритм k-середніх будує k кластерів, розташованих на можливо великих відстанях один від одного. Основний тип задач, які вирішує алгоритм k-середніх – наявність припущень (гіпотез) щодо числа кластерів, при цьому вони повинні бути різні настільки, наскільки це можливо. Вибір числа k може базуватися на результатах попередніх досліджень, теоретичних міркуваннях або інтуїції.

Загальна ідея алгоритму: задана фіксована кількість k кластерів спостереження зіставляються кластерам так, що середні в кластері (для всіх змінних) максимально можливо відрізняються один від одного.

Опис алгоритму.

1. Початковий розподіл об'єктів по кластерах.

Вибирається число k, і на першому кроці ці точки вважаються «центрами» кластерів. Кожному кластеру відповідає один центр.

Вибір початкових центроїдів може здійснюватися в такий спосіб:

- o вибір k-спостережень для максимізації початкової відстані;
- o випадковий вибір k-спостережень;
- o вибір перших k-спостережень.

В результаті кожен об'єкт призначений певного кластеру.

2. Ітеративний процес.

Обчислюються центри кластерів, якими потім і далі вважаються покоординатно середні кластерів. Об'єкти знову перерозподіляються. Процес обчислення центрів і перерозподілу об'єктів триває до тих пір, поки не виконана одна з умов:

- кластерні центри стабілізувалися, тобто всі спостереження належать кластеру, до якого належали до поточної ітерації;
- кількість ітерацій дорівнює максимальній кількості ітерацій.

Вибір кількості кластерів є складним питанням. Якщо немає припущень щодо цієї кількості, рекомендують створити 2 кластера, потім 3, 4, 5 і т.д., порівнюючи отримані результати.

Перевірка якості кластеризації.

Після отримань результатів кластерного аналізу методом k -середніх слід перевірити правильність кластеризації (тобто оцінити, наскільки кластери відрізняються один від одного). Для цього розраховуються середні значення для кожного кластера. При вірній кластеризації повинні бути отримані середні, які сильно відрізняються для всіх вимірювань або хоча б більшої їх частини.

Переваги алгоритму k -середніх:

- простота використання;
- швидкість використання;
- зрозумілість і прозорість алгоритму.

Недоліки алгоритму k -середніх:

- алгоритм занадто чутливий до викидів, які можуть спотворювати середнє. Можливим вирішенням цієї проблеми є використання модифікації алгоритму – алгоритм k -медіани;
- алгоритм може повільно працювати на великих базах даних. Можливим вирішенням цієї проблеми є використання вибірки даних.

ЗАВДАННЯ ДЛЯ ПЕРЕВІРКИ ЗНАНЬ

1. Як виконується графічне представлення статистичної оцінки даних?
2. Що таке машинне навчання?
3. В чому полягає ансамблевий метод машинного навчання бустінг?
4. Що відносять до мір центральної тенденції при виконанні статистичної оцінки даних?
5. Що таке класифікація у машинному навчанні?
6. В чому полягає принцип роботи рекурентної нейромережі?
7. Що відносять до мір мінливості при виконанні статистичної оцінки даних?
8. Що таке регресія у машинному навчанні?

9. В чому полягає суть нейромережевого машинного навчання?
10. Як виконується перевірки дослідів на відтворюваність?
11. Що таке кластеризація у машинному навчанні?
12. В чому полягає принцип роботи згорткової нейромережі (CNN)?
13. Що таке матриця планування експерименту?
14. Що таке узагальнення у машинному навчанні?
15. Для чого виконують аналіз предметної області при традиційному процесі аналізу даних?
16. Що таке стандартне відхилення, як воно визначається?
17. Що таке пошук правил у машинному навчанні?
18. Для чого виконують постановку задачі при традиційному процесі аналізу даних?
19. Які є найважливіші напрямки, на яких зосереджуються для вирішення питань управління даними?
20. Де використовують навчання з підкріпленням?
21. Що включає у себе етап підготовки даних при традиційному процесі аналізу даних?
22. В чому полягає суть алгоритму обробки великої кількості інформації MapReduce?
23. Що є метою навчання з підкріпленням?
24. Що визначає якість даних (Data quality) при традиційному процесі аналізу даних?
25. Що таке GFS при розробці рішень в області Big Data?
26. В чому полягає ансамблевий метод машинного навчання стекінг?
27. Що необхідно виконати інженеру у традиційному програмуванні, щоб отримати рішення?
28. У чому відмінність NoSQL від SQL БД?
29. В чому полягає ансамблевий метод машинного навчання беггінг?
30. Що необхідно виконати інженеру у машинному навчанні, щоб отримати рішення?

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Your Guide to Data Science Careers URL: <https://www.coursera.org/articles/data-science-career> (дата звернення: 05.06.2024).
2. Encyclopedia of Data Science and Machine Learning. USA, IGI Global, 2023. 3143 p.

3. Introducing Data Science Big Data, Machine Learning, and More, Using Python Tools. URL: https://www.academia.edu/43139523/Introducing_Data_Science_Big_Data_Machine_Learning_and_more_using_Python_tools?ri_id=2008 (дата звернення: 05.06.2024).

4. Introducing Data Science. URL: <https://www.scribd.com/document/318279227/Introducing-Data-Science> (дата звернення: 05.06.2024).

5. Introduction to DATA SCIENCE. A PYTHON Path for a Non-computer Scientist. URL: https://www.academia.edu/110868688/Introduction_to_DATA_SCIENCE_A_PYTHON_Path_for_a_Non_computer_Scientist (дата звернення: 05.06.2024).

6. Data Science Bookcamp: Five Real-world Python Projects. URL: <https://www.manning.com/books/data-science-bookcamp> (дата звернення: 05.06.2024).

7. Гороховатський В. О., Творошенко І. С. Методи інтелектуального аналізу та оброблення даних: навч. посібник. Харків: ХНУРЕ, 2021. 92 с.

МЕТОДИ ІМІТАЦІЙНОГО МОДЕЛЮВАННЯ ТА ОПТИМІЗАЦІЇ (укладач Гуменюк П. О.)

ТЕМА 1 ВИДИ І МЕТОДИ ІМІТАЦІЙНОГО МОДЕЛЮВАННЯ [2-4]

1.1 Способи дослідження систем

У певні моменти функціонування більшості систем виникає необхідність їх дослідження з метою отримання уявлення про внутрішні відношення між їх компонентами або обчислення їх продуктивності в нових умовах експлуатації. Ці дослідження можна проводити різними способами. Розглянемо їх детальніше.

– *Експеримент із реальною системою чи з моделлю системи?* Якщо можливо фізично змінити систему (якщо це рентабельно) і запустити її в дію в нових умовах, найкраще вчинити саме так, оскільки в цьому випадку питання про адекватність отриманого результату зникає сам собою. Однак часто такий підхід нездійснений або через занадто великі витрати на його виконання, або в силу руйнівного впливу на саму систему.

– *Якщо експеримент з моделлю, то фізична чи математична модель?* В деяких випадках створення фізичних моделей може виявитися дуже ефективним для дослідження технічних систем або систем управління. Однак переважна більшість створюваних моделей є математичними. Вони представляють систему засобами логічних і кількісних відношень, які потім піддаються обробці та змінам, щоб визначити, як система реагує на зміни, точніше – як би вона реагувала, якби існувала насправді.

– *Якщо математична модель, то аналітичне рішення чи імітаційне моделювання?* Якщо модель досить проста, можна обчислити її співвідношення та параметри і отримати точне аналітичне рішення. Однак деякі аналітичні рішення можуть бути надзвичайно складними і вимагати величезних комп'ютерних ресурсів. У цьому випадку модель слід вивчати за допомогою імітаційного моделювання, тобто багаторазового випробування моделі з потрібними вхідними даними, щоб визначити їх вплив на вихідні критерії оцінки роботи системи.

Імітаційні моделі класифікують за трьома аспектами.

– *Статична чи динамічна?* Статична імітаційна модель – це система в певний момент часу або система, в якій час просто не відіграє ніякої ролі. Динамічна імітаційна модель представляє систему, що змінюється в часі, наприклад конвеєрну систему на заводі.

– *Детермінована чи стохастична?* Якщо імітаційна модель не містить ймовірнісних (випадкових) компонентів, вона називається детермінованою. Стохастичні імітаційні моделі видають результат, який є випадковим сам по собі, і

тому він може розглядатися лише як оцінка істинних характеристик моделі.

– *Неперервна чи дискретна?* Дискретна модель не завжди використовується для моделювання дискретної системи, і навпаки. Чи необхідно для конкретної системи використовувати дискретну чи неперервну модель, залежить від задач дослідження.

Комп'ютерне моделювання – це застосування комп'ютерних технологій розв'язання математичних моделей на електронно-обчислювальних машинах.

Комп'ютерні моделі складних систем поділяються умовно на такі види:

- структурно-функціональні, які представляють собою умовний образ об'єкта, описаний за допомогою програмних та комп'ютерних технологій;
- імітаційні, що являють собою програму або комплекс програм, що дозволяє відтворювати процеси функціонування об'єкта в різних умовах;
- комбіновані, з можливостями спостереження та дослідження об'єкта на динамічних умовних образах моделі та імітаційних моделях об'єкта.

Існує багато програмних комплексів, які дозволяють проводити побудову та дослідження моделей (моделювання). Кожне програмне середовище має свій інструментарій та дозволяє працювати з певними видами інформаційних моделей.

Від вибору програмного середовища залежить алгоритм побудови комп'ютерної моделі, і навіть форма його представлення.

1.2 Види імітаційного моделювання

Імітаційне моделювання застосовується, коли неможливо побудувати аналітичну модель системи, що враховує причинні зв'язки, наслідки, нелінійності, стохастичні змінні, коли необхідно імітувати поведінку системи в часі, розглядаючи різні можливі сценарії її розвитку за зміни зовнішніх і внутрішніх умов.

Таким чином, *імітаційне моделювання* – це високорівнева інформаційна технологія із застосуванням комп'ютерів, що найчастіше використовується при моделюванні складних систем.

Переваги імітаційного моделювання:

- висока адекватність між фізичною суттю описуваного процесу і його моделлю;
- можливість описати складну систему на досить високому рівні деталізації;
- значно більше областей дослідження, ніж аналітичне моделювання;
- відсутність обмежень відображення в моделі залежностей між параметрами моделі;
- можливість оцінки функціонування системи не тільки в стаціонарних станах, але і в перехідних режимах (процесах);

– одержання значної кількості даних про досліджуваний об'єкт (закон розподілу випадкових величин, числові значення абсолютні та відносні, і багато іншого);

– найбільш раціональне ставлення «результат-витрати» по відношенню до аналітичного і фізичного моделюванню.

Недоліки імітаційного моделювання:

– розробка хорошої моделі часто обходиться дорожче, ніж аналітична і вимагає більше часу на створення і налагодження;

– складно оцінити ступінь точності моделі, її адекватність досліджуваному процесу;

– відносно високі вимоги до кваліфікації дослідника для написання моделі;

– спільність застосування та індивідуальність реалізації.

Імітаційне моделювання умовно може бути представлене різними різновидами чи напрямками, що відповідно мають свої методології (рис. 1.1). Розглянемо ці напрямки.

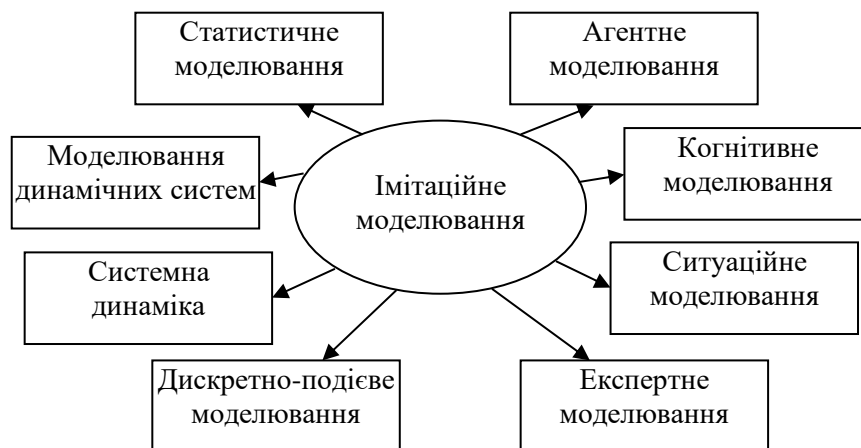


Рисунок 1.1 – Різновиди імітаційного моделювання

1.3 Методи імітаційного моделювання

В імітаційному моделюванні під методом мається на увазі якась основа, яку ми використовуємо, щоб «перевести» систему з реального світу у світ моделей. Метод передбачає певну мову, «положення та умови» для розробки моделі. На даний момент існує три методи (рис. 1.2):

- системна динаміка;
- дискретно-подієве моделювання;
- агентне моделювання.

Кожен метод застосовується у певному діапазоні рівнів абстракції.

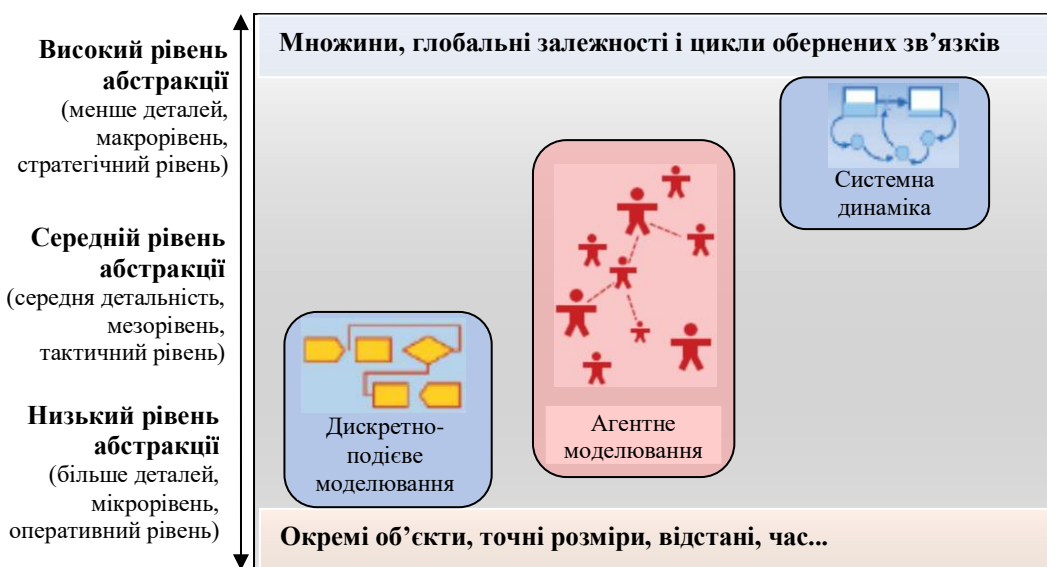


Рисунок 1.2 – Рівні абстракції видів моделювання

Системна динаміка передбачає дуже високий рівень абстракції і, як правило, застосовується для стратегічного моделювання.

Дискретно-подієве моделювання підтримує середній та низький рівні абстракції.

Між ними знаходяться *агентні моделі*, які можуть бути як дуже деталізованими, коли агенти представляють фізичні об'єкти, так і гранично абстрактними, коли за допомогою агентів моделюються конкуруючі компанії або уряди держав.

Імітаційні моделі, які ми будемо розглядати, будуть в більшості дискретними, динамічними та стохастичними. Будемо їх називати дискретно-подієвими моделями.

1.4 Статистичне моделювання

Статистична модель випадкового процесу – це алгоритм, за допомогою якого імітують роботу складної системи, схильної до випадкових збурень; імітують взаємодію елементів системи, що носять імовірнісний характер.

Чисельний метод, який вирішує задачу генерування послідовності випадкових чисел із заданими законами розподілу, отримав назву «метод статистичних випробувань» або «метод Монте-Карло».

Метод статистичного імітаційного моделювання – це спосіб вивчення складних процесів і систем, що піддаються випадковим збуренням, за допомогою імітаційних моделей.

Статистичне моделювання залежно від області застосування поділяється на кілька напрямків (рис. 1.3).



Рисунок 1.3 – Класифікація видів статистичного чисельного моделювання за типом розв'язуваних задач

Ймовірнісне моделювання (методи Монте-Карло) – цей напрям розвивається як спосіб розв'язання математичних задач – обчислення інтегралів, розв'язання систем лінійних рівнянь та диференціальних рівнянь.

Ймовірнісно-імітаційне моделювання – застосування теорії ймовірності та методів Монте-Карло для побудови імітаційних моделей у молекулярній, статистичній, квантовій, нейтронній фізиці, геофізиці, газовій динаміці, хімічній кінетиці, у передачі та захисті інформації, у моделях масового обслуговування, фінансовій математиці, біології та ін.

Статистичне моделювання – застосування математичної статистики для статистичного оцінювання та прогнозування, кореляційно-регресійного та багатовимірного статистичного аналізу, оптимізації систем, визначення екстремуму функцій великої кількості змінних та ін. у різних галузях виробництва та науки. Результатом застосування статистичного моделювання є можливість проведення прогнозу.

1.5 Моделювання динамічних систем

Система – це сукупність об'єктів, що функціонують і взаємодіють один з одним для досягнення певної мети. Насправді поняття системи залежить від задач конкретного дослідження.

Стан системи визначається сукупністю змінних, необхідних для опису системи на певний момент часу відповідно до задач дослідження.

Існують системи двох типів: дискретні та неперервні.

У *дискретній* системі змінні стану в різні періоди часу змінюються миттєво.

У *неперервній* системі змінні змінюються неперервно в часі.

Під *динамічною системою* будемо розуміти будь-який об'єкт, процес або явище, для якого однозначно визначене поняття стану як сукупності деяких величин і заданий закон, який описує зміну початкового стану з плином часу, який рухається в просторі і змінюється в часі.

В основі методології моделювання динамічних систем і побудови об'єктно-

орієнтованих моделей в технічних системах лежить агрегативний підхід, де складна система представлялася у вигляді агрегату (чорної скриньки), що має множини вхідних, вихідних і впливаючих керуючих сигналів. Цей підхід широко використовується при дослідженні складних індивідуальних управлінських систем, до яких належать АСУ. Агрегативні системи дозволяють описати широке коло об'єктів дослідження з відображенням системного характеру цих складних об'єктів, з можливістю розбиття складної системи на скінченну кількість підсистем, зі збереженням зв'язків між ними та взаємодії частин. В теорії автоматичного управління основним об'єктом вивчення є системи керування складними динамічними (технічними) об'єктами та їх елементами.

Для моделювання динамічних систем використовуються так звані середовища схемотехнічного моделювання: VISSIM, SIMULINK + MATLAB, PowerSim, Multisim, LabView, Easy5, MvStudium та ін.

1.6 Системна динаміка

Системна динаміка – вид моделювання, де для досліджуваної системи будуються графічні діаграми причинних зв'язків і глобальних впливів одних параметрів на інші в часі, а потім створена на основі цих діаграм модель імітується на комп'ютері.

Системна динаміка представляє методи вивчення динамічних систем. Передбачається, що користувач:

- моделює систему як закриту структуру, яка сама визначає власну поведінку;
- виявляє цикли оберненого зв'язку врівноважуючого чи підсилюючого типу. Цикли оберненого зв'язку займають центральне місце в системній динаміці;
- встановлює накопичувачі та потоки, які на них впливають.

Накопичувачі характеризують стан системи. Вони містять пам'ять системи. Модель працює тільки із сукупністю об'єктів: окремі елементи, що містяться в накопичувачі, не помітні.

Потоки представляють інтенсивність, з якою змінюються стани системи.

Системна динаміка найчастіше використовується для розробки довгострокових стратегічних моделей і передбачає високий рівень агрегації об'єктів: моделі системної динаміки розглядають людей, товари, ресурси та інші окремі елементи в кількісних термінах.

1.7 Дискретно-подієве моделювання

Дискретно-подієве моделювання – підхід до моделювання, що пропонує абстрагуватися від неперервної природи подій і розглядати тільки основні події

модельованої системи, такі як: «очікування», «обробка замовлення», «рух з вантажем», «розвантаження» та інші.

Дискретно-подієве моделювання використовується для побудови моделі, що відображає розвиток системи в часі, коли стани змінних змінюються миттєво у конкретні моменти часу.

У такі моменти часу відбуваються події, при цьому *подія* визначається як миттєве виникнення, що може змінити стан системи.

Дискретно-подієве моделювання передбачає представлення модельованої системи у вигляді процесу, тобто послідовності операцій, що виконуються з агентами.

Модель задається графічно у вигляді діаграми процесу, блоки якої є окремими операціями. Як правило, діаграма процесу починається з блоку «джерело», що генерує агентів. Цей блок передає агентів у наступні блоки діаграми, що задають операції модельованого процесу. Завершується діаграма процесу зазвичай блоком, що знищує цих агентів.

Як час прибуття агентів, так і час їх обслуговування зазвичай є випадковими величинами, і їх значення генеруються функціями розподілу ймовірностей. Тому і самі дискретно-подієві моделі є стохастичними, і для отримання репрезентативного результату модель повинна пропрацювати певний час, або ж потрібно виконати певну кількість прогонів моделі.

Даний підхід широко застосовується в теорії масового обслуговування, яка вивчає широкий клас випадкових процесів у системах поширення інформації, інформаційно-комунікативних системах та в різних галузях масового обслуговування.

1.8 Агентне моделювання

Агентне моделювання (agent-based model (ABM)) – різновид імітаційного моделювання, метод, який досліджує поведінку децентралізованих агентів і те, як така поведінка визначає поведінку всієї системи в цілому.

Агент – деяка сутність, що має активність, автономну поведінку, може приймати рішення відповідно до деяких наборів правил, взаємодіяти з оточенням, а також самостійно змінюватися.

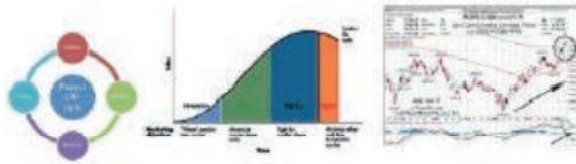
Агентами можуть бути різні об'єкти (рис. 1.4): транспортні засоби, обладнання, проекти, організації, земельні ділянки, люди і т.д.

Агентне моделювання включає елементи теорії ігор, складних систем, мультиагентних систем та еволюційного програмування, методи Монте-Карло, використовує випадкові числа.

Люди в різних ролях:
покупці, співробітники, пацієнти,
лікарі, клієнти, солдати...



Нематеріальні речі:
проекти, продукти, інновації, ідеї,
інвестиції...



Обладнання, транспорт:
вантажівки, машини, крани, літаки,
вагони, верстати...



Організації:
компанії, політичні партії, країни...



Рисунок 1.4 – Агентні об'єкти

Середовище – деякий простір, в якому знаходяться агенти, який характеризується своїми станами і факторами, агенти знаходяться у певному місці цього простору, з можливістю орієнтування та пересування в даному просторі.

Правила взаємодії – закони взаємодії агентів у навколишньому середовищі, з процедурами ухвалення рішення та вибору стратегії при черговому кроці взаємодії.

Задачі імітаційного моделювання при агентному підході полягають у визначенні характеристик стану агентів і середовища, вивчення поведінки агентів при різних ситуаціях взаємодії і зміні станів середовища.

Через вивчення поведінки множини агентів у певному середовищі відповідно до деяких правил взаємодії можливе прогнозування поведінки системи в цілому

Мета агентних моделей – отримати уявлення про ці глобальні правила, загальну поведінку системи, виходячи з припущень про індивідуальну поведінку її окремих активних об'єктів і про взаємодію цих об'єктів в системі.

На даний момент не існує стандартної мови агентного моделювання. Структура агентної моделі може бути визначена як графічно, так і за допомогою сценаріїв. Поведінка агента може бути задана різними способами. Якщо у агента є стан, від якого залежать його дії та реакції, то його поведінку найкраще задавати за допомогою діаграми станів. Іноді поведінка агента задається діями, що виконуються при настанні певних подій.

1.9 Когнітивне моделювання

Під когнітивними технологіями розуміється широкий спектр технологій раціоналізації та формалізації інтелектуальних систем створення та функціонування знань, експертизи, комунікації та прийняття рішення. Когнітивні інформаційні технології являють собою сукупність методів, прийомів, дій, процесів, що здійснюються в певній послідовності, інструментальних засобів (ПК), що дозволяють перетворити вхідну інформацію у варіанти управлінського рішення.

В даний час когнітивний підхід є напрямом дослідження великих систем, до яких належать соціально-економічні, політичні, екологічні системи. Методологія когнітивного моделювання розвивається у напрямку вдосконалення апарату аналізу, моделювання та пошуку рішень у слабоформалізованих та погано структурованих ситуаціях за відсутності або неповної інформації про процеси, що відбуваються в таких ситуаціях та умовах швидких змін.

1.10 Ситуаційне моделювання

Ситуаційне моделювання – метод дослідження ситуацій, що включає в себе побудову моделі реальної ситуації та проведення з нею різноманітних уявних експериментів: прогнозування напрямів її розвитку та (або) «програвання» на ній гаданих рішень з управління ситуацією з метою вибору оптимального. З метою дослідження ситуацій різного виду розробляють комп'ютерні аналітичні та імітаційні моделі для прикладних застосувань.

1.11 Області застосування імітаційного моделювання

Імітаційне моделювання довело свою успішність у багатьох сферах застосування. Поява нових методів моделювання та зростання обчислювальної потужності комп'ютерів дозволяє стверджувати, що кількість цих областей лише зростатиме.

На рисунку 1.5 представлено розподіл областей застосування імітаційного моделювання відповідно до рівнів абстракції, що використовуються в моделях.

У нижній частині малюнка розташовуються моделі фізичного рівня, де об'єкти реального світу моделюються максимально докладно. На цьому рівні враховується фізична взаємодія, розміри, швидкість, відстань.

Моделі, розташовані у верхній частині схеми, абстрактніші і найчастіше оперують узагальненими поняттями, такими як сукупність споживачів або статистика рівня зайнятості, а не окремими об'єктами.

Інші моделі мають середній рівень абстракції.

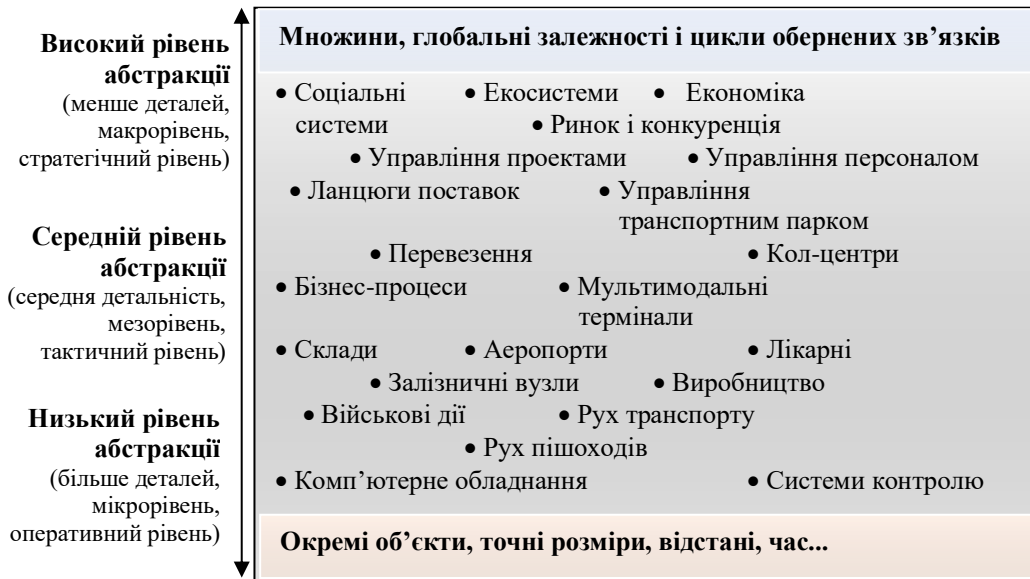


Рисунок 1.5 – Застосування імітаційного моделювання

Основні переваги імітаційного моделювання.

1. Імітаційні моделі дозволяють аналізувати системи і знаходити рішення в тих випадках, коли такі методи, як аналітичні обчислення та лінійне програмування не справляються із завданням.

2. Після визначення рівня абстракції, розробляти імітаційну модель буде набагато простіше, ніж аналітичну, оскільки процес створення моделі буде модульним.

3. Структура імітаційної моделі природним чином відображає структуру системи, що моделюється.

4. Імітаційна модель дозволяє відстежувати всі об'єкти системи, враховані в обраному рівні абстракції, додавати дані та проводити статистичний аналіз.

5. Однією з головних переваг імітаційного моделювання є можливість програвати модель у часі та анімувати її поведінку. Анімація буде незаперечною перевагою при демонстрації моделі і може виявитися корисною для верифікації моделі та знаходження помилок.

6. Імітаційні моделі набагато переконливіші за електронні таблиці.

ТЕМА 2 ЕТАПИ РОЗРОБКИ МОДЕЛЕЙ [1,4,5]

2.1 Етапи моделювання

Моделювання – від постановки задачі до отримання результатів проходить такі етапи.

I. Аналіз вимог та проектування.

1. Постановка та аналіз задачі та мети моделювання.
2. Збір та аналіз вихідної інформації про об'єкт моделювання.
3. Побудова концептуальної моделі.
4. Перевірка достовірності концептуальної моделі.

II. Розробка моделі.

1. Вибір середовища моделювання.
2. Упорядкування логічної моделі.
3. Призначення властивостей модулів моделі.
4. Задання модельного часу.
5. Верифікація моделі.

III. Проведення експерименту.

1. Запуск моделі, прогін моделі.
2. Варіювання параметрів моделі та збір статистики.
3. Аналіз результатів моделювання.

IV. Підбиття підсумків моделювання згідно з поставленою метою та задачею моделювання.

На першому етапі моделювання – «Аналіз вимог та проектування» – формулюється концептуальна модель, будується її формальна схема та вирішується питання про ефективність та доцільність моделювання системи.

Концептуальна модель (КМ) – це абстрактна модель, що визначає склад і структуру системи, властивості елементів та причинно-наслідкові зв'язки, що властиві аналізованій системі та суттєві для досягнення цілей моделювання.

На другому етапі моделювання – «Розробка моделі» – відбувається уточнення або вибір програмного пакета моделювання. Програмні та технічні засоби вибираються з урахуванням ряду критеріїв. Неодмінна умова при цьому – достатність і повнота засобів реалізації концептуальної моделі. Серед інших критеріїв можна назвати доступність, простоту та легкість освоєння, швидкість та коректність створення програмної моделі.

Для того, щоб виконати підетап «Завдання модельного часу» введемо поняття модельного часу.

В імітаційній моделі змінна, що забезпечує поточне значення модельного часу, називається *годинником модельного часу*. Існує два основних підходи до просування модельного часу: *рух часу від події до події* та *рух часу з постійним кроком*.

Підхід, що використовує *просування часу в моделі від події до події*, застосовується всіма основними імітаційними програмами та більшістю розробників, які створюють свої моделі універсальними мовами. На рисунку 2.1 вигнуті стрілки ілюструють просування годинника модельного часу.

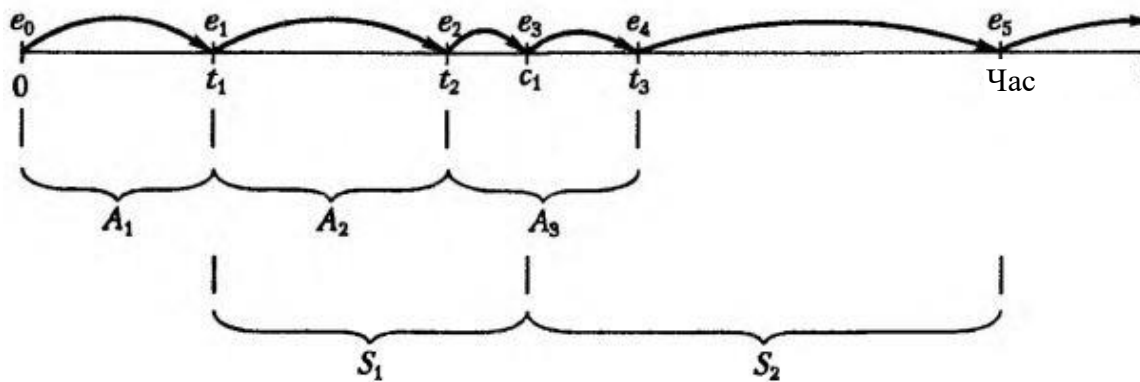


Рисунок 2.1 – Механізм просування часу від події до події

В момент часу $e_0=0$ пристрій перебуває у стані незайнятості. Час надходження першої заявки визначається шляхом генерації значення A_1 і додавання його до 0 . Після цього годинник модельного часу переходить на час виникнення найближчої події, і в цей момент оновлюється стан системи з урахуванням події, що відбулася, та інформації про час виникнення майбутніх подій. Потім годинник модельного часу просувається до часу виникнення наступної (нової) найближчої події, оновлюється стан системи і визначається час майбутніх подій, і т. д. Процес просування модельного часу від часу виникнення однієї події до часу виникнення іншої триває доти, доки не буде виконана будь-яка умова зупинки, вказана заздалегідь. Слід зазначити, що тривалість інтервалу руху модельного часу від однієї події до іншої може бути різною.

При просуванні часу з постійним кроком Δt годинник модельного часу просуваються точно на Δt одиниць часу для будь-якого відповідного вибору значення Δt . Після кожного оновлення годинника виконується перевірка, щоб визначити, чи відбулися якісь події протягом попереднього інтервалу часу Δt чи ні. Якщо на цей інтервал заплановано одну або кілька подій, вважається, що ці події відбуваються наприкінці інтервалу, після чого стан системи та статистичні лічильники відповідним чином оновлюються. Просування часу у вигляді постійного кроку показано на рисунку 2.2, де вигнуті стрілки показують просування годинника модельного часу, а e_i ($i=1,2,\dots$) – це дійсний час виникнення події i будь-якого типу, а не значення годинника модельного часу.

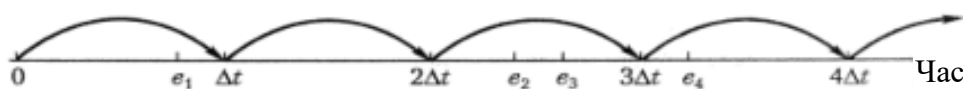


Рисунок 2.2 – Приклад просування модельного часу за допомогою постійного кроку

На інтервалі $[0, \Delta t]$ подія відбувається в момент часу e_1 , але вона розглядається як подія в момент часу Δt . На інтервалі $[\Delta t, 2\Delta t]$ події не відбуваються, але все ж таки модель виконує перевірку, щоб переконатися в цьому. На інтервалі $[2\Delta t, 3\Delta t]$ події відбуваються в моменти часу e_2 і e_3 , проте вважається, що вони сталися в момент часу $3\Delta t$ і т. д. Таким чином, просування часу за допомогою постійного кроку має два недоліки: виникнення помилок, пов'язаних з обробкою подій наприкінці інтервалу, протягом якого вони відбуваються, а також необхідність вирішувати, яку подію обробляти першою, якщо події, що насправді відбуваються в різний час, розглядаються як одночасні.

Зважаючи на цю обставину, просування часу за допомогою постійного кроку не використовують у дискретно-подієвих імітаційних моделях, коли інтервали часу між послідовними подіями можуть значно відрізнятись за своєю тривалістю.

Третій етап – «Проведення експерименту» – є вирішальним, у якому, завдяки процесу імітації модельованої системи, відбувається збирання необхідної інформації, її статична обробка в інтерпретації результатів моделювання, у результаті приймається рішення: або дослідження буде продовжено, або завершено. Якщо відомий результат, можна порівняти його з отриманим результатом моделювання. Якщо результати не відповідають цілям моделювання (реальному об'єкту або процесу), значить, допущені помилки на попередніх етапах або вхідні дані не є коректними, тому розробник повертається до одного з попередніх етапів.

Підетап «*Аналіз результатів моделювання*» представляє собою всебічний аналіз одержаних результатів з метою отримання рекомендацій по проектуванню системи або її модифікації.

На *четвертому етапі* «Підбиття підсумків моделювання згідно поставленої мети та завдання моделювання» проводять оцінку виконаної роботи, зіставляють поставлені цілі з отриманими результатами та створюють остаточний звіт щодо виконаної роботи.

2.2 Компоненти дискретно-подієвої імітаційної моделі та їх організація

Дискретно-подієва імітаційна модель, яка використовує механізм просування часу від події до події і написана універсальною мовою, містить наступні компоненти:

- *стан системи* – сукупність змінних стану, необхідних для опису системи у певний час;
- *годинник модельного часу* – змінна, що вказує на поточне значення модельного часу;
- *список подій* – список, що містить час виникнення кожного наступного

типу подій;

- *статистичні лічильники* – змінні, призначені для зберігання статистичної інформації про характеристику системи;

- *програма ініціалізації* – підпрограма, що встановлює імітаційну модель у початковий стан в момент часу, рівний 0;

- *синхронізуюча програма* – підпрограма, яка відшукує наступну подію у списку подій і потім переводить годинник модельного часу на час виникнення цієї події;

- *програма обробки подій* – підпрограма, яка оновлює стан системи, коли відбувається подія визначеного типу (для кожного типу подій існує окрема програма обробки подій);

- *бібліотечні програми* – набір підпрограм, що застосовуються для генерації випадкових спостережень з розподілу ймовірностей, які були визначені як частина імітаційної моделі;

- *генератор звітів* – підпрограма, яка зчитує оцінки (зі статистичних лічильників) критеріїв оцінки роботи і видає звіт по закінченні моделювання;

- *основна програма* – підпрограма, яка викликає синхронізуючу програму для того, щоб визначити наступну подію, а потім передає управління відповідній подієвій програмі з метою забезпечення заданого оновлення стану системи.

Логічні зв'язки (потік управління) між вказаними компонентами показані на рисунку 2.3.

Як уже зазначалось, система – це чітко визначена сукупність об'єктів. Об'єкти характеризуються за допомогою значень, іменованих *атрибутами*. У дискретно-подієвій імітаційній моделі ці атрибути є частиною стану системи. Об'єкти з будь-якою загальною властивістю часто об'єднуються у *списки* (у *файли*, або *набори*). Для кожного об'єкта існує запис у списку, що складається з атрибутів об'єктів.

Організація та функціонування дискретно-подієвої моделюючої програми, в якій застосовується механізм просування часу від події до події, досить типові. Такий підхід до імітаційного моделювання називається *плануванням подій*, оскільки час майбутніх подій явно вказаний у моделі та запланований у модельному майбутньому.

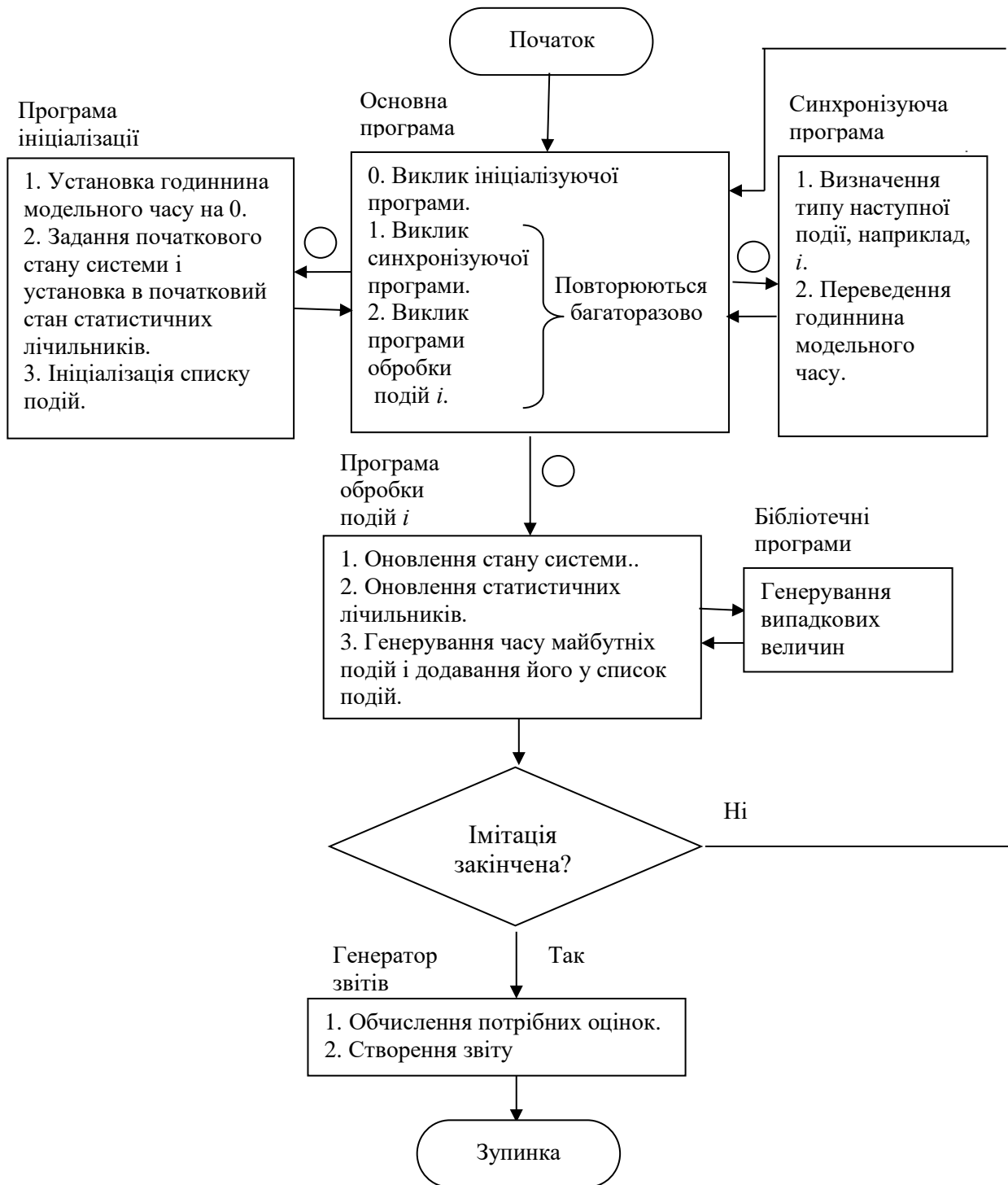


Рисунок 2.3 – Потік керування у механізмі просування часу від події до події

В той же час існує альтернативний підхід до імітаційного моделювання, що називається *процесним підходом*.

При цьому моделювання розглядається з точки зору окремих об'єктів, що беруть участь у ньому, і розроблений код описує «досвід» окремого «типового» об'єкта в міру його «переміщення» по системі.

Схема процесу, що відповідає надходженню об'єкта в систему та його обслуговування єдиним пристроєм, показано на рисунку 2.4.

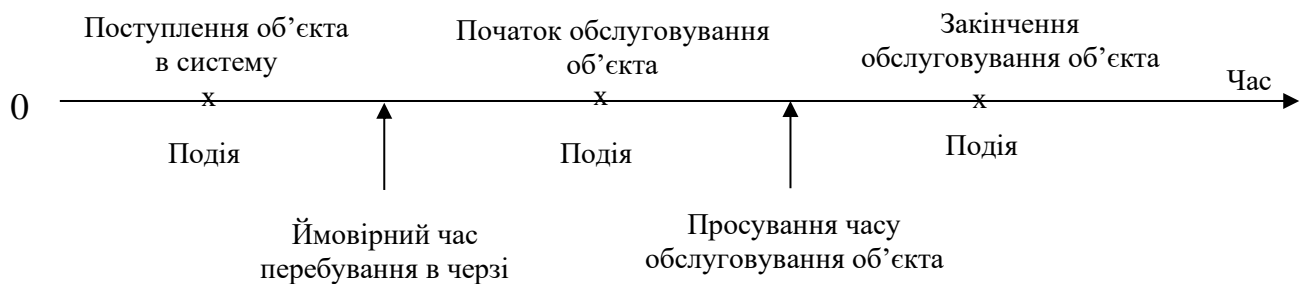


Рисунок 2.4 – Схема процесу просування об'єкта у системі.

У системі чи імітаційній моделі може бути кілька різних процесів. Кожному з них відповідає програма, в якій описується вся «історія існування» об'єкта в міру його переміщення в даному процесі. Програма процесу забезпечує просування модельного часу. На відміну від програми обробки подій у програмі процесу може бути кілька точок входу.

Першою подією у списку подій є поступлення об'єкта в систему. Запис для надходження наступного об'єкта міститься у переліку подій. Наступний об'єкт надійде в момент часу, рівний часу надходження поточного об'єкта плюс час між надходженнями. Якщо пристрій зайнятий, об'єкт поміщається в кінець черги і чекає, поки він не буде обраний пристроєм обслуговування в деякий момент часу в модельному майбутньому. Така ситуація називається *умовним очікуванням*. Якщо пристрій вільний, обслуговування об'єкта починається негайно.

2.3 Методи дослідження імітаційних моделей

Модель системи призначена для отримання інформації про властивості реального об'єкта. Ця інформація накопичується в результаті проведення ряду експериментів. При цьому «експеримент» представляє собою виконання комп'ютерної імітаційної моделі.

У термінології планування експериментів вхідні змінні та структурні припущення, що становлять модель, називаються *факторами*, а вихідні показники роботи – *відгуками*.

Фактори можуть бути або *кількісними*, або *якісними*. Кількісні фактори, як правило, припускають числові значення, тоді як якісні фактори зазвичай представляють собою структурні припущення, які не вимірюються кількісно.

При проведенні імітаційних експериментів розрізняють *керовані* і *не керовані фактори* залежно від того, чи може управління відповідними реальними

системами керувати ними, чи ні. В імітаційному моделюванні доступне управління всіма факторами незалежно від їх практичної керованості.

При моделюванні перед виконанням прогонів моделі *планування експериментів* дозволяє вирішити, які саме конфігурації слід створювати, щоб отримати потрібну інформацію при найменшому обсязі моделювання.

Спеціальні засоби планування експериментів дозволяють скласти план проведення експериментів, що забезпечує збір необхідної для отримання обґрунтованих висновків інформації, при мінімальних витратах.

На першому етапі планування дослідник задає:

- кількість варійованих факторів;
- кількість рівнів кожного фактора (рівнями називають ті кількісні або якісні значення фактора, при яких можливо проводити експеримент);
- область зміни факторів;
- необхідну точність та довірчу ймовірність вимірювання відгуку моделі.

Кількість рівнів найчастіше приймають рівним двом, що відповідає припущенню про лінійну залежність відгуку моделі від факторів.

На другому етапі приступають до *тактичного* планування експерименту, метою якого є визначення таких умов експерименту, що забезпечать вимірювання відгуку моделі з заданою точністю та довірчою ймовірністю. В результаті цього етапу планування повинні бути визначені тривалість одного прогону (в одиницях модельного часу) і кількість прогонів.

На третьому етапі приступають до *стратегічного* планування експериментів, метою якого є визначення такої серії експериментів, що забезпечить отримання бажаної інформації про відгук моделі при мінімальних затратах. В результаті цього етапу планування потрібно визначити тип аналізу, який буде використаний, та побудувати матрицю планування відповідно до обраного типу аналізу та заданої кількості факторів.

За матрицею планування проводиться серія експериментів і за їх результатами будується рівняння регресії.

ТЕМА 3 ІМІТАЦІЙНЕ МОДЕЛЮВАННЯ СИСТЕМ [1,4,5]

3.1 Переваги та недоліки моделювання систем

Імітаційне моделювання широко застосовується у світі на дослідження складних систем. Цьому сприяють переваги, властиві цьому методу.

1. Більшість складних реальних систем з ймовірнісними параметрами не можна точно описати з використанням математичних моделей.

2. Шляхом моделювання можна розробити ряд альтернативних варіантів

моделей системи і потім визначити, який найбільш відповідає вхідним вимогам.

3. Імітаційне моделювання часом набагато менш затратне, ніж проведення експериментів із реальними системами. Тим більше що іноді експерименти на реальних системах в принципі неможливі.

4. Моделювання дозволяє вивчити тривалий інтервал функціонування системи у стислі терміни чи, навпаки, вивчити докладніше роботу системи у розгорнутий інтервал часу.

5. При динамічному імітаційному моделюванні можна отримувати будь-яку кількість оцінок імовірнісної моделі, проводячи її прогони. Детальне вивчення одержаних оцінок прийнятно використовувати при оптимізації моделі.

Найбільш відомими і найбільш поширеними засобами формалізації процесів функціонування дискретних систем є мережі масового обслуговування та мережі Петрі.

Мережі масового обслуговування призначені для опису процесів обробки, таких як виробничі процеси, бізнес-процеси, і представляють великий, але чітко обмежений клас систем.

Мережі Петрі є більш могутнім засобом формалізації дискретних процесів і описують системи із найскладнішими зв'язками, в тому числі управляючими.

3.2 Системи масового обслуговування

Основні поняття.

Система масового обслуговування (СМО) – це система, яка обслуговує вимоги, що надходять до неї (заявки).

Вимоги (заявки) на обслуговування надходять через дискретні (постійні або випадкові) інтервали часу.

Канали – прилади, необхідні для обслуговування цих заявок. Обслуговування триває деякий час, постійний або випадковий

Мережі масового обслуговування функціонують за наступними простими правилами. Вимоги, що надходять на обслуговування, проходять одну за одною СМО згідно вказаного маршруту. На вході кожної СМО вимога намагається знайти один вільний пристрій з декількох паралельно функціонуючих. Правило, за яким вимога обирає той чи інший пристрій за умови, що вони вільні, може бути вказано окремо відповідно до процесу функціонування реальної системи. Якщо всі пристрої однакові, то вимога переглядає пристрої по порядку і займає перший, що виявився вільним. Якщо в момент надходження вимоги до СМО всі пристрої виявились зайнятими, то вимога при наявності черги у СМО намагається в неї потрапити, або залишає мережу масового обслуговування і вважається не обслуженою.

Процес роботи СМО представляє собою випадковий процес з дискретними станами та неперервним часом. Стан СМО змінюється стрибком в моменти реалізації подій (надходження нової або закінчення обслуговування вимоги, моменту, коли вимога, виходить з черги). З вимог, які вже обслуговані, формується вихідний потік. Кожна СМО, залежно від кількості каналів, їх продуктивності, характеру потоку заявок, має деяку пропускну здатність, яка дозволяє їй більш чи менш успішно справлятися з потоком вимог. Задача теорії масового обслуговування полягає в побудові моделей, які пов'язують задані умови роботи СМО з показниками ефективності системи, що описують її спроможність впоратися з потоком вимог. Під ефективністю обслуговуючої системи розуміють характеристику рівня виконання цією системою функцій, для яких вона призначена.

Класифікація систем масового обслуговування.

Системи масового обслуговування класифікують за різними ознаками.

За наявності черг:

1 системи з *відмовами* (з втратами) – заявка, яка надійшла в момент, коли всі канали зайняті, отримує відмову, покидає СМО і надалі не обслуговується.

2 системи з *очікуванням* (без втрат) – заявка, що прийшла в момент, коли всі канали зайняті, не відкидається, а стає в чергу і чекає можливості бути обслугованою. Системи з очікуванням у свою чергу класифікують:

а) *по довжині черги* – з обмеженою довжиною черги, які допускають чергу, але з обмеженою кількістю місць в ній;

б) *за часом очікування* – з обмеженим часом очікування, що допускають чергу, але з обмеженим терміном перебування кожної вимоги в ній;

с) *по дисципліні обслуговування* – з обслуговуванням по пріоритету, що допускають чергу, але деякі заявки обслуговуються поза чергою (тобто по пріоритету).

За кількістю каналів обслуговування:

1 *одноканальні*;

2 *багатоканальні*.

За кількістю заявок, які можуть одночасно перебувати в обслуговуючій системі:

1 *з обмеженим потоком вимог*;

2 *з необмеженим потоком вимог*.

За розташуванням обробних пристроїв:

1 *впорядковані* (чергова вимога надходить спочатку на перший пристрій і лише в тому випадку, якщо він зайнята, передається на другий ...);

2 *невпорядковані* (всі інші системи обслуговування, в яких заявки

розподіляються між обробними пристроями за будь-яким іншим принципом).

За місцем знаходження джерела вимог:

1 *розімкнені*, коли заявки надходять ззовні мережі і після обробки залишають її;

2 *замкнуті*, коли деяка кількість заявок весь час знаходиться в ній, переходячи з однієї СМО до іншої, але ніколи не залишаючи мережу МО.

Елементи мережі масового обслуговування.

При графічному представленні мережі МО користуються позначеннями, представленими на схемі (рис. 3.1).

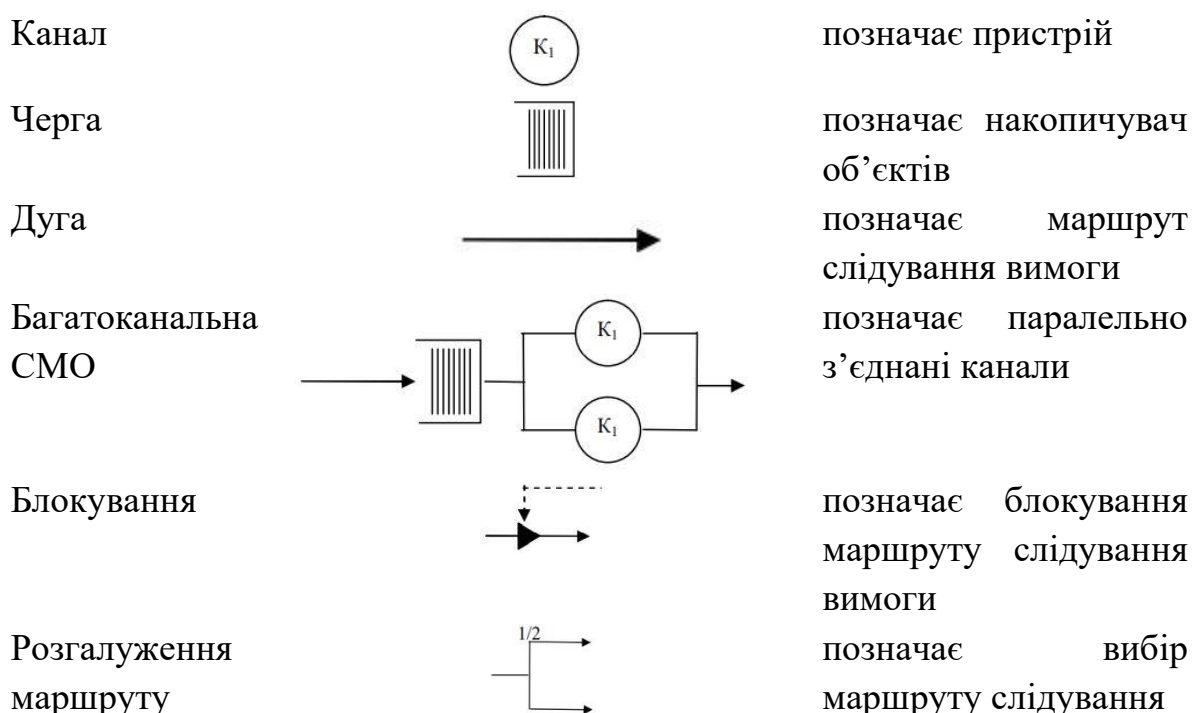


Рисунок 3.1 – Позначення графічного представлення мережі МО

Розглянемо, наприклад, функціонування мережі МО, яка представлена на рисунку 3.2.

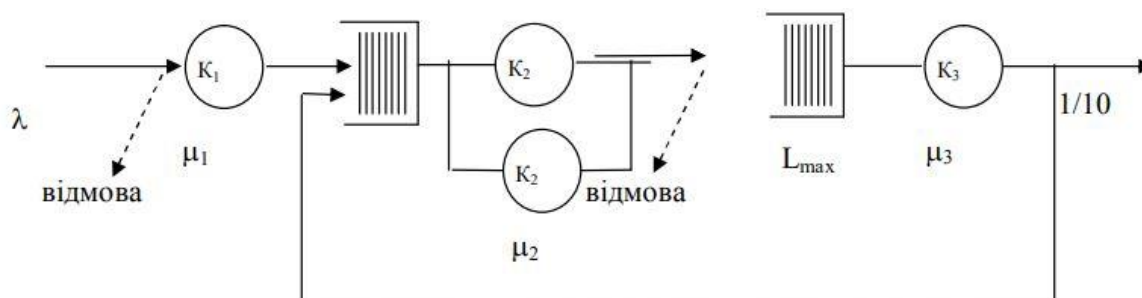


Рисунок 3.2 – Приклад мережі масового обслуговування

Усі об'єкти проходять три етапи обробки – СМО₁, СМО₂, СМО₃. До першої СМО об'єкти надходять на обслуговування з інтенсивністю λ (вимог/од.часу), де обслуговуються пристроєм K_1 з інтенсивністю μ_1 (вимог/од.часу). Оскільки в першій СМО відсутня черга, то об'єкти, що надійшли в момент часу, коли пристрій зайнятий, отримують відмову в обслуговуванні і вважаються не обслуженими мережею МО. Ті об'єкти, що пройшли обслуговування у першому пристрої мережі МО, надходять до другої СМО. У другій СМО об'єкти обслуговуються в одному з каналів обслуговування K_2 . Якщо в момент надходження об'єкта до другої СМО обидва канали зайняті, об'єкт надходить до черги другої СМО. Обмеження на довжину черги другої СМО не вказане, тому кількість об'єктів, що одночасно знаходяться в цій черзі, може бути будь-якою великою. Коли закінчилось обслуговування вимоги в одному з каналів другої СМО, об'єкт намагається потрапити до третьої СМО. Об'єкт потрапляє на обслуговування в канал третьої СМО, якщо він вільний, в противному випадку він намагається зайняти місце в черзі третьої СМО. Але черга третьої СМО має обмежену кількість місць L_{\max} . Це означає, що об'єкт, який намагається потрапити до черги третьої СМО у момент, коли всі місця в ній зайняті, отримує відмову в обслуговуванні.

Об'єкт, який закінчив обслуговування у каналі третьої СМО, з ймовірністю $1/10$ повертається до другої СМО для додаткової обробки і з ймовірністю $9/10$ виходить з мережі МО як успішно обслугований. Такі повернення на додаткову обробку зустрічаються, наприклад, у системах з пунктами контролю якості: з ймовірністю $9/10$ об'єкт проходить перевірку якості успішно і з ймовірністю $1/10$ в результаті перевірки виявляється, що об'єкт неякісний і потребує додаткової обробки.

Всім процесам масового обслуговування як простим, так і складним, притаманна одна характерна риса, за якою дослідник приймає рішення про можливість використання мережі масового обслуговування для моделювання досліджуваного процесу: в процесі масового обслуговування вимога не знищується, не з'єднується з іншими вимогами і не розпадається на декілька вимог.

Для того, щоб представити систему засобами мережі масового обслуговування потрібно:

- з'ясувати, що являється в системі об'єктом обслуговування;
- виділити елементи процесу обслуговування об'єктів і кожному елементу поставити у відповідність СМО;
- для кожної СМО визначити кількість пристроїв та наявність черги;
- з'єднати СМО у відповідності до процесу обслуговування;
- визначити маршрут проходження об'єкту обслуговування від однієї СМО до іншої;

- визначити умови надходження в кожную СМО (ймовірність вибору маршруту та інші);
- визначити наявність блокування маршруту та умови блокування;
- визначити числові значення параметрів кожної СМО;
- визначити числові значення параметрів зовнішнього потоку на обслуговування;
- визначити стан мережі масового обслуговування на початку моделювання.

Мережі Петрі.

Мережі Петрі є засобом формального опису процесів функціонування дискретних систем. У дискретній системі зміни її стану трапляються в особливі моменти часу, коли виникають умови для здійснення події. Здійснення події означає зміну стану системи, а значить, виникнення або не виникнення умов для інших подій. Процес функціонування дискретної системи – це упорядкована в часі послідовність подій.

Структура мережі Петрі задається спрямованим двостороннім графом, що складається з позицій і переходів. Основні елементи мережі Петрі представлені на схемі (рис. 3.3).

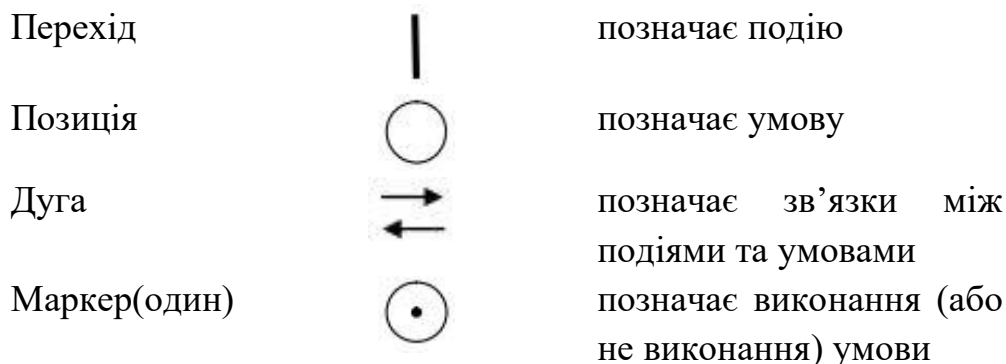


Рисунок 3.3 – Основні елементи мережі Петрі

Події зображуються планками, які називають *переходами*. Умови для виникнення подій зображуються кругами, які називають *позиціями*. Зв'язки між позиціями та переходами зображуються дугами. Виконання (або не виконання) умови зображується наявністю (або відсутністю) *маркера* в позиції. Маркер позначається крапкою.

Переходи в мережі Петрі є подіями, які змінюють стан реальної системи. На рисунку 3.4 наведено приклад інтерпретації мережі Петрі.

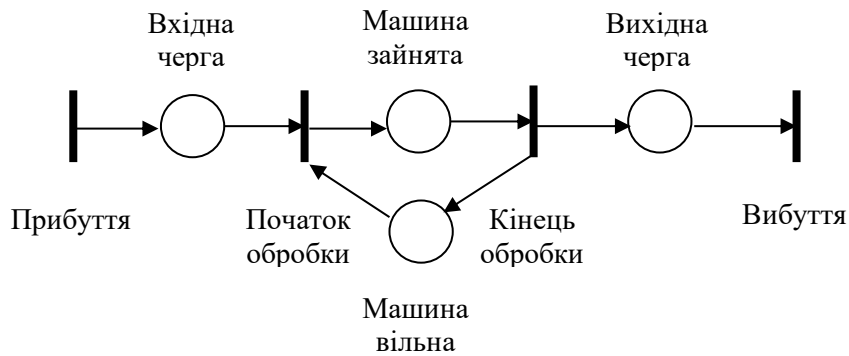


Рисунок 3.4 – Інтерпретація мережі Петрі

Мережа Петрі виконується за допомогою запусків переходів у результаті видалення фішок з його вхідних позицій та додавання в вихідні позиції переходу. Процес продовжується, доки запускається хоч один перехід, або, доки вичерпаний час спостереження мережі Петрі. Послідовність спрацьовувань переходів повністю визначає поведінку мережі. Таким чином, мережа Петрі описує структуру системи, її стан та поведінку.

ТЕМА 4 СТАТИСТИЧНЕ МОДЕЛЮВАННЯ. МЕТОД МОНТЕ-КАРЛО [6-8]

4.1 Суть методу Монте-Карло

Методи Монте-Карло (ММК) – група чисельних методів вивчення випадкових процесів. Суть методу полягає в наступному: процес описується математичною моделлю з використанням генератора випадкових величин, модель багаторазово обраховується, на основі отриманих даних обчислюються ймовірнісні характеристики процесу, що розглядається.

Великою перевагою методу Монте-Карло є те, що він дозволяє врахувати у моделі елемент випадковості та складність реального світу. Крім того, метод є робастним по надійним до зміни різних параметрів, таких як розподіл випадкової величини. У основі лежить закон великих чисел.

Одним із типових прикладів використання методу Монте-Карло є задачі, в яких необхідно знайти математичне очікування деякої випадкової величини. Для цього необхідно згенерувати набір випадкових значень цієї величини і визначити середнє. Випадкова величина зазвичай характеризується певним розподілом ймовірностей.

Суть методу Монте-Карло полягає в наступному: для цільової випадкової величини генерується набір випадкових значень, а потім на його основі розраховуються необхідні значення.

Для прогнозування ризиків, прибутковості, термінів окупності та інших

фінансових результатів використовується метод Монте-Карло-симуляції. Ймовірність події визначається так: програма вибирає комбінації випадкових значень (наприклад, несприятливих результатів) і виходячи з цього видає усереднений результат. Для знаходження точнішого значення симуляцію слід повторити кілька разів.

Програмне забезпечення застосовується різне – від знайомого усім Excel до вузькоспеціалізованих програмних продуктів, які використовуються фінансовими аналітиками, фізиками, програмістами, трейдерами та ін.

4.2. Схема методу Монте Карло

Імітаційне моделювання за методом Монте-Карло представляє собою визначення математичного очікування (середнього значення випадкової величини) шляхом проведення певної кількості симуляцій (випробувань).

Тобто, потрібно знайти значення математичного очікування A деякої величини, що вивчається. Для цього вибирають таку випадкову величину X , математичне очікування якої дорівнює A : $M(X) = A$. Практично ж поступають так: проводять N випробувань, в результаті яких отримують N можливих значень X , обчислюють їхнє середнє арифметичне і приймають його як оцінку (наближене значення) шуканого числа A .

Спрощено схему алгоритму можна представити у наступному вигляді (рис 4.1).

Перший етап – розробка прогнозної моделі, передбачає формування очікуваної імітаційної моделі, яка повинна адекватно відображати майбутній сценарій реалізації проєкту.

Другий етап – виявлення чинників ризику, включає відбір ключових змінних для моделювання.

Третій етап – визначення умов кореляції, полягає у встановленні формальної залежності між результативним показником і відібраними ключовими змінними.

Четвертий етап – імовірнісний розподіл відібраних ключових змінних, передбачає здійснення таких кроків:

- 1) визначення обмежень можливої зміни відібраних ключових змінних;
- 2) встановлення ймовірнісної ваги за межами значень.

П'ятий етап – імітаційне прогнозування, вимагає генерування випадкових сценаріїв реалізації проєкту з використанням вибраних допущень.

Шостий етап – аналіз отриманих результатів, потребує здійснення статистичної оцінки та інтерпретації одержаних результатів імітації.

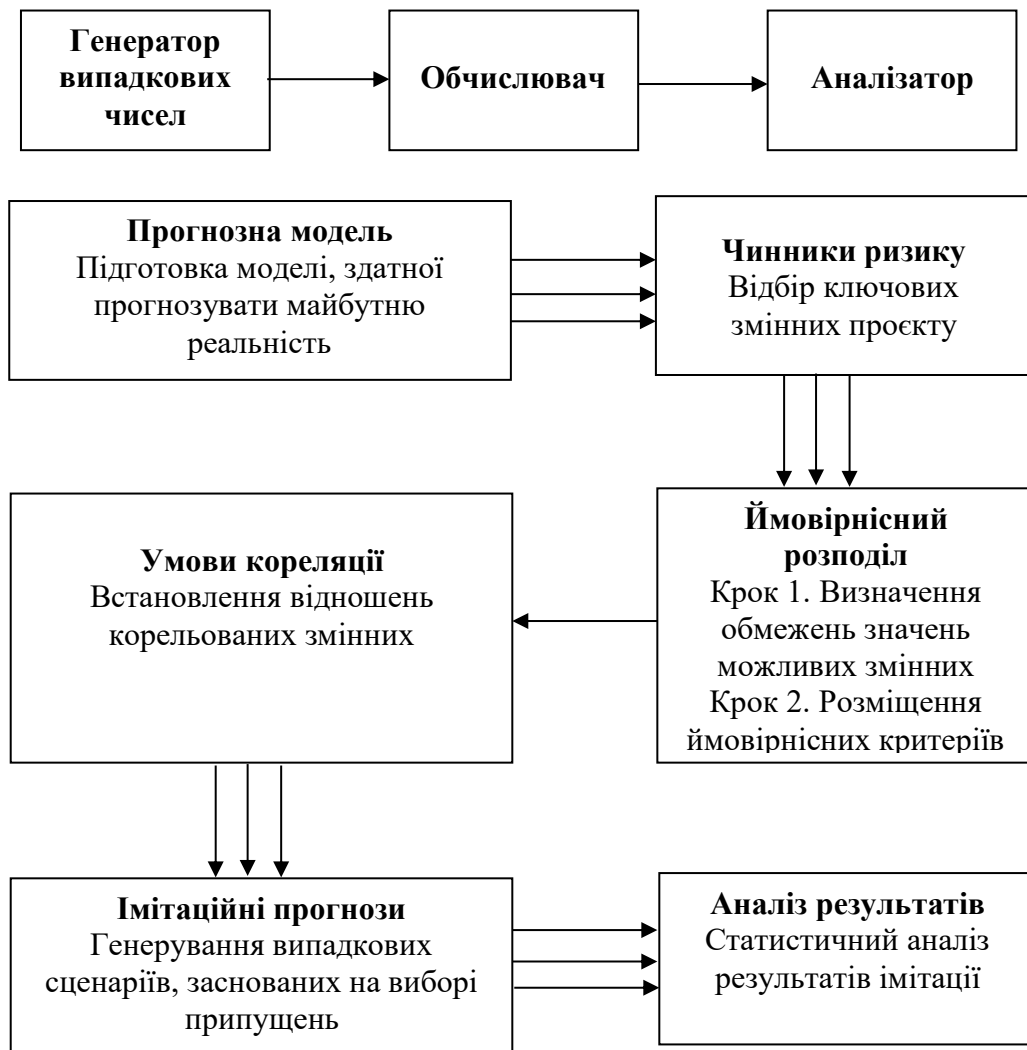


Рисунок 4.1 – Спрощена схема алгоритму Монте-Карло.

Вхідні дані.

Дані для отримання необхідної величини визначаються шляхом стохастичної (випадкової) вибірки. Щоб було зрозуміліше, наведемо найпростіший приклад із комп'ютерних ігор.

Припустимо, у нас є комп'ютерна гра, в яку ми грали багато разів. При цьому ведеться статистика: зіграно 100 ігор, із них 30 перемог, 70 поразок. Це буде вхідними даними. А рішення буде таким: ймовірність перемоги – 30 %, програшу – 70 %.

Процес моделювання методом Монте-Карло.

Під час створення моделі всі невизначені чинники замінюються діапазоном можливих значень. Наприклад, жоден аналітик, який займається оцінкою ризиків, не може знати, яким буде курс євро через 3-5 років. Програма дозволяє встановити діапазон значень на розсуд фахівця.

Далі система розподіляє ймовірність. Для оцінки різних параметрів

застосовуються варіанти розподілу:

Нормальний розподіл.

Крива нормального розподілу або крива Гаусса виглядає так (рис. 4.2).



Рисунок 4.2 – Крива Гаусса

Значення випадкової величини, розташоване посередині, характеризує найвищу ймовірність. Для побудови кривої використовуються статистичні дані: очікуване значення та стандартне відхилення. Такий варіант розподілу підійде, наприклад, до прогнозування розрахунку вартості комунальних послуг у майбутньому.

Рівномірний розподіл.

Усі події можуть наступити з однаковою ймовірністю, користувачеві потрібно лише встановити мінімум та максимум. Крива рівномірного розподілу має вигляд прямокутника. На графіку (рис. 4.3) a та b – мінімальні значення, C – ймовірність виникнення події. Підійде до розрахунку умовно-постійних витрат у короткостроковому періоді.

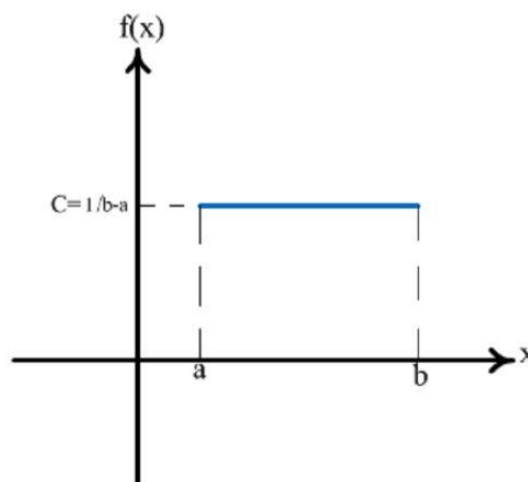


Рисунок 4.3 – Крива рівномірного розподілу

Логнормальний розподіл.

Використовується для розрахунку параметрів, які не можуть набувати від'ємного значення, але можуть зростати до нескінченності. Приклад: ціни на енергоносії. Графік логнормального розподілу має несиметричний вигляд (рис. 4.4).

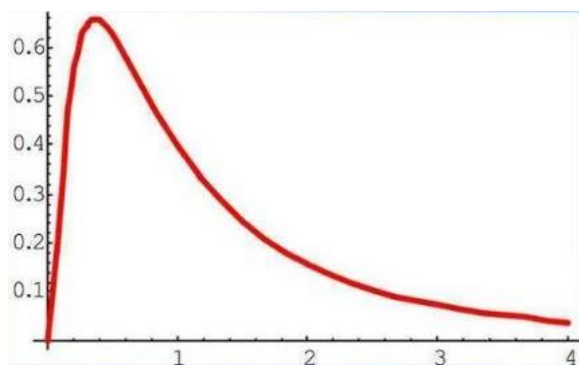


Рисунок 4.4 – Крива логнормального розподілу

Дискретний розподіл.

Може застосовуватися, наприклад, для комп'ютерних ігор, де обчислюється відсоток перемог та поразок. Користувачеві потрібно обчислити потрібні значення у складі можливих.

Трикутний розподіл.

Визначаються точки мінімуму, максимуму та максимальної ймовірності (рис. 4.5). Значення, близькі до точки максимальної ймовірності, приймаються як можливі. Цей варіант застосовується для аналізу продажів, котирувань цінних паперів, прогнозу фінансових результатів за підсумками діяльності та ін. Трикутний розподіл підходить для імітаційного моделювання ризиків методом Монте-Карло.

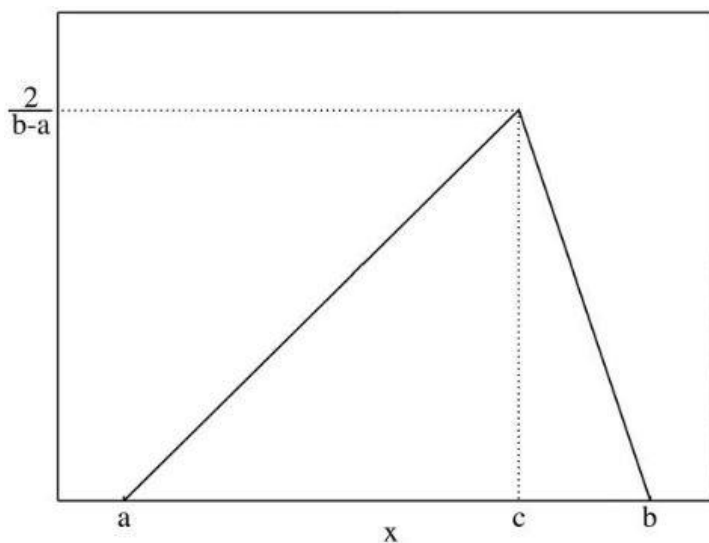


Рисунок 4.5 – Крива трикутного розподілу

Таким чином, імітаційне моделювання методом Монте-Карло виконується багаторазово. За результатами всіх операцій отримується вибірка значень, результати систематизуються і визначається підсумкова ймовірність події.

Вихідні дані.

Вихідними або підсумковими даними імітаційного моделювання за методом Монте-Карло можуть бути числові значення чи відсотки. В окремих випадках значення можуть знаходитись увсередині діапазону.

4.3 Переваги та недоліки методу Монте-Карло

Перевагами методу Монте-Карло є:

- простота та універсальність – метод може застосовуватися практично до будь-якого типу даних;
- метод дозволяє враховувати як певний тип даних окремо, так і взаємозв'язки між різними типами даних;
- метод можна застосовувати там, де не спрацьовують звичні методи дослідження, засновані на математичних розрахунках.

Недоліки методу Монте-Карло:

- іноді потрібне проведення великої кількості випробувань, що може тривати багато часу;
- для виконання симуляцій методом Монте-Карло у програмі необхідно залучати кваліфікованих фахівців;
- метод не може дати достовірну оцінку для подій, що характеризуються дуже низькою чи дуже високою ймовірністю настання.

4.4 Генератори випадкових величин

Способи генерування випадкових величин. Імітаційні моделі складних систем містять випадкові величини, що мають різні закони розподілу. При побудові алгоритму імітації ці випадкові величини реалізуються генераторами випадкових чисел. Від якості генераторів випадкових чисел, що використовуються, залежить точність результатів імітаційного моделювання.

Відомі наступні способи генерування випадкових величин:

- зберігання у комп'ютері таблиці випадкових чисел і отримання потім з неї даних для імітаційного моделювання;
- використання деякого фізичного пристрою, наприклад електронної лампи, для генерації випадкового шуму;
- застосування рекурсивних формул коли на підставі i -того випадкового числа обчислюється $i+1$ -ше випадкове число.

Недоліком першого способу є зберігання великого обсягу інформації та

повільна швидкість. Недоліком другого способу – неможливість направленою експерименту з параметрами моделі. Третій спосіб не має недоліків попередніх способів і в теперішній час є найбільш прийнятним.

Випадкові величини моделюються на ЕОМ з допомогою програмних давачів випадкових величин. Давач випадкових величин – це програма, яка видає на запит одне випадкове значення випадкової величини $z \in \{0 \leq t \leq 1\}$. Шляхом багаторазового звернення до давача випадкової величини отримують вибірку незалежних випадкових значень.

Генератори випадкових чисел, що рівномірно розподілені на інтервалі (0; 1), зазвичай є вбудованими в програмне забезпечення. Тому досліднику, як правило, не доводиться самостійно будувати такі генератори.

ТЕМА 5 ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ІМІТАЦІЙНОГО МОДЕЛЮВАННЯ [4]

Одне з найбільш важливих рішень, які доводиться приймати розробникам моделей або аналітикам, стосується вибору програмного забезпечення. Якщо програмне забезпечення недостатньо гнучке або з ним складно працювати, то імітаційне моделювання може дати неправильні результати або виявитися взагалі невиконуваним.

Використання *пакета імітаційного моделювання* порівняно із застосуванням універсальної мови програмування дає кілька переваг.

– Пакети імітаційного моделювання автоматично надають більшість функціональних можливостей, потрібних для створення імітаційної моделі, що дозволяє істотно скоротити час, необхідний для програмування, і загальну вартість проєкту.

– Пакети імітаційного моделювання забезпечують природне середовище для створення імітаційних моделей. Їх основні моделюючі конструкції, більше підходять для імітаційного моделювання, ніж відповідні конструкції в універсальних мовах програмування.

– Імітаційні моделі, які створені за допомогою пакетів моделювання, як правило, простіше модифікувати та використовувати.

– Пакети імітаційного моделювання забезпечують досконаліші механізми виявлення помилок, оскільки вони виконують автоматичний пошук помилок багатьох типів.

Тим не менше, деякі імітаційні моделі (особливо пов'язані з оборонною сферою), як і раніше, створюються за допомогою *універсальних мов* моделювання,

які теж мають деякі переваги.

- Мови програмування знає більшість розробників, чого не можна сказати про пакети імітаційного моделювання.

- Швидкість виконання прогону імітаційних моделей, написаних універсальними мовами, зазвичай вища, ніж моделей, створених за допомогою пакетів імітаційного моделювання (це пов'язано з тим, що такі пакети часто розробляються для найрізноманітніших систем за допомогою одного набору моделей конструкцій, тоді як програма може бути успішніше написана під конкретну систему). Однак з появою недорогих швидкодіючих персональних комп'ютерів ця перевага дещо втратила свою актуальність.

- При програмуванні універсальні мови забезпечують більшу гнучкість, ніж пакети імітаційного моделювання.

- Вартість застосовуваного програмного забезпечення зазвичай нижча.

5.1 Класифікація програмних засобів імітаційного моделювання

Історично пакети імітаційного моделювання розділилися на два основні типи: мови імітаційного моделювання та предметно-орієнтовані програми моделювання.

Мови імітаційного моделювання за своєю природою універсальні, розробка моделі тут має на увазі написання коду. У цілому вони надають досить гнучкі можливості моделювання, але часто складні у використанні.

Програми моделювання орієнтовані на вирішення певної задачі, в них модель розробляється з використанням графіки, діалогових вікон і меню, що розкриваються. Програми моделювання іноді простіше вивчати і використовувати, але при вирішенні деяких завдань вони не можуть забезпечити достатню гнучкість моделювання.

Таким чином, існує два типи пакетів імітаційного моделювання. *Універсальні пакети імітаційного моделювання* призначені для різних цілей, але вони можуть мати спеціальні функції для вирішення одного конкретного виду задач (наприклад, моделювання виробничих систем, систем зв'язку або модернізації технологій виробництва).

Предметно-орієнтовані пакети імітаційного моделювання служать для вирішення спеціальних задач, таких, як моделювання роботи виробничих систем, медичних закладів, центрів виконання замовлень.

5.2 Можливості програм імітаційного моделювання

При виборі програмних засобів імітаційного моделювання слід враховувати всі можливості, що надаються ними, які можна об'єднати в наступні групи:

- основні характеристики;
- сумісність обладнання та програмного забезпечення;
- анімація;
- засоби отримання та обробки статистичних даних;
- послуги, що надаються замовникам та документація;
- звіти з вихідними даними та графіками

Найважливішою властивістю, яку повинен мати програмний продукт імітаційного моделювання, є *гнучкість при моделюванні*, тобто можливість моделювати системи з різним рівнем складності технологічних операцій.

Наступною важливою властивістю засобів імітаційного моделювання є *простота в застосуванні* (і вивченні), тому багато сучасних пакетів моделювання забезпечені графічним інтерфейсом користувача.

Для складних систем може виявитися корисним ієрархічне моделювання. *Ієрархія* дозволяє згрупувати кілька конструктивних основних елементів моделювання в нові структурні компоненти вищого рівня. Ці нові структурні компоненти потім можна об'єднати в структурні компоненти ще вищого рівня і т. д.

При моделюванні багатьох систем дуже важливою є висока *швидкість роботи* моделі.

Ще одна корисна можливість полягає у *збереженні стану моделі в кінці прогону* і використанні збережених даних для того, щоб пізніше відновити моделювання.

Нарешті, при купівлі програмних засобів імітаційного моделювання велике значення має їхня *вартість*.

При виборі програмного засобу імітаційного моделювання необхідно звертати увагу на те, для якої *комп'ютерної платформи* воно призначене. Майже всі програмні засоби працюють на персональних комп'ютерах із системою Windows, деякі працюють на робочих станціях UNIX та комп'ютерах Apple. Якщо пакет може працювати на різних платформах, тоді він має забезпечувати *сумісність із іншими платформами*. Крім того, слід врахувати необхідний *обсяг оперативної пам'яті*. Потрібно також пам'ятати про те, *які операційні системи підтримує пакет*.

Наявність у багатьох пакетах вбудованих засобів анімації – одна з причин зростання інтересу до імітаційного моделювання. При використанні *анімації* всі ключові елементи системи представлені на екрані піктограмами, які динамічно змінюють своє положення, колір і форму в процесі розвитку імітаційної моделі в часі.

Існує *два основних типи анімації*: спільна і роздільна (також називається

відтворенням).

Спільна анімація здійснюється під час прогону імітаційної моделі. Але треба мати на увазі, що зазвичай при виконанні робочих прогонів анімація відключається, так як вона уповільнює їх виконання.

При *роздільній анімації* зміни стану системи зберігаються у файлі на диску і застосовуються для керування графікою *після* завершення моделювання. У деяких програмних засобах імітаційного моделювання реалізовано обидва типи анімації.

До засобів анімації також висуваються певні **вимоги**.

Перш за все, необхідно, щоб програмне забезпечення імітаційного моделювання використовувало *анімацію за умовчанням* як частину процесу створення моделі.

Оскільки в першу чергу засоби анімації необхідні для візуалізації роботи системи та обміну інформацією, повинна існувати можливість створення піктограм з *високою роздільною здатністю* та збереження їх для подальшого застосування.

Крім того, потрібно, щоб програмні середовища були оснащені *бібліотекою стандартних піктограм* і забезпечували плавний рух піктограм.

Також повинен існувати елемент керування для *прискорення або уповільнення* ефектів анімації.

Необхідна й можливість *збільшення або зменшення масштабу* зображення, а також *прокручування* зображення для перегляду різних частин системи, якщо вона повністю не поміщається на екрані.

Багато, щоб для анімації застосовувалася *векторна графіка, а не растрова*. Перший вид графіки забезпечує обертання об'єктів (скажімо, несучого гвинта вертольота) і збереження напрямку руху (наприклад, транспортного засобу, що здійснює поворот).

Деякі пакети імітаційного моделювання забезпечують *об'ємну (тривимірну) анімацію*. У пакетах з тривимірною анімацією можна надати глядачеві можливість переміщатися по системі на певному об'єкті.

Повинна існувати можливість *застосування в анімації імпортованих CAD-креслень та графічних об'єктів Clip Art*.

Часто може бути корисним виведення на екран *динамічної графіки та статичних даних* по мірі виконання моделювання. В якості динамічної графіки зазвичай використовуються годинник, кругова шкала, індикатор рівня (що відображає, наприклад, чергу) і динамічно оновлювані гістограми і графіки часової залежності.

Якщо імітаційна модель не забезпечена надійними засобами статистичного аналізу, неможливо буде отримувати достовірні дані про роботу модельованої системи. Насамперед у програмі необхідний хороший *генератор випадкових чисел*,

тобто механізм для генерування незалежних значень, рівномірно розподілених в інтервалі $[0,1]$.

В цілому кожен фактор випадковості системи повинен бути представлений в імітаційній моделі *розподілом ймовірностей*, а не тільки *середнім значенням*. Якщо можна підібрати *стандартний теоретичний розподіл*, що дозволяє оптимально представити певний фактор випадковості, такий розподіл слід використовувати в моделі системи.

Програмне забезпечення повинне підтримувати наступні *неперервні розподіли*: експоненційний, Вейбулла, логарифмічно нормальний, нормальний, рівномірний і трикутний, а також гамма- і бета-розподіл.

Крім неперервних мають бути доступні ще й *дискретні розподіли*: біноміальний, геометричний, а також розподіл Пуассона і дискретно-рівномірний.

Якщо немає можливості знайти теоретичний розподіл, що вдало представляє певний фактор випадковості, треба скористатися *емпіричним* (або *визначеним користувачем*) розподілом, який ґрунтується на отриманих даних. При цьому випадкові числа отримують шляхом вибірки значень випадкової величини з функцією розподілу, створеною за спостережуваними системними даними.

Програма повинна містити команду, яка дозволяє виконувати *незалежні повторення* (реплікації) чи *прогони* імітаційної моделі.

Програмні засоби імітаційного моделювання повинні надавати можливість вказувати, *для оцінювання яких критеріїв роботи системи будуть збиратись вихідні дані*, а не просто видавати відомості, що обраховуються за замовчуванням і не представляють для користувача ніякого інтересу.

Важливим є також питання *оптимізації*. Метою оптимізації є виконати прогони імітаційної моделі за допомогою інтелектуальних модулів оптимізації і знайти в результаті комбінацію шуканих змінних, що дають оптимальне або близьке до нього рішення. У таких модулях оптимізації застосовуються такі евристичні засоби: генетичні алгоритми, нейронні мережі, а також розсіюючий пошук і забороняючий пошук.

Для успішної роботи з програмним забезпеченням необхідна якісна документація. Користувач повинен мати можливість вивчити пакет імітаційного моделювання, не звертаючись до спеціальних курсів.

Для оцінки показників роботи моделі в програмі повинна бути закладена можливість створювати *стандартні звіти*.

Деякі програми імітаційного моделювання дозволяють виводити результати у вигляді *стовпчикових або кругових діаграм*. Крім того, бажано, щоб *результати для різних модельованих сценаріїв зберігалися в базі даних* і при необхідності могли бути зображені на одному графіку.

Нарешті, корисним для визначення залежності у вихідних даних, що отримуються в результаті одного прогону імітаційної моделі, може бути *графік кореляційної функції*.

Повинна також існувати можливість експортувати результати спостережень виходів окремої моделі у такі програмні пакети, як електронні таблиці, бази даних, статистичні та графічні пакети, з метою проведення подальшого аналізу.

ЗАВДАННЯ ДЛЯ ПЕРЕВІРКИ ЗНАНЬ

Сформулюйте розширені відповіді на поставлені питання і перевірте їх повноту та правильність за допомогою матеріалів літературних джерел.

1. Розгляньте кроки, які виконуються під час дослідження складних систем та розробки імітаційних моделей. Дайте визначення терміну «адекватність моделі реальним об'єктам». З'ясуйте, що означає оцінювання адекватності моделі.

2. Наведіть приклади задач, що їх можна розв'язати за допомогою методу машинної імітації. Складіть алгоритм розв'язання задачі з використанням методу однорідного градування системного часу та методу неоднорідного градування системного часу.

3. Назвіть мови програмування, в яких використовується однорідне або неоднорідне градування системного часу.

4. Розгляньте приклади проблемно-орієнтованих, машинно-орієнтованих, спеціалізованих мов моделювання. З'ясуйте, у чому полягають переваги та вади кожного типу мов програмування під час їх використання для реалізації імітаційних моделей. Розгляньте, на які класи поділяють спеціалізовані мови імітаційного моделювання та приклади мов моделювання серед кожного класу спеціалізованих мов.

5. За допомогою літературних джерел поясніть, за якими ознаками розрізняють мови імітаційного моделювання.

6. Наведіть приклади застосування імітаційних моделей під час вивчення діючої функціональної системи.

7. Наведіть приклади застосування імітаційних моделей під час аналізу гіпотетичної функціональної системи.

8. Наведіть приклади застосування імітаційних моделей під час проектування досконалішої системи.

9. З'ясуйте, у чому полягають переваги та вади кожного типу мов програмування під час їх використання для реалізації імітаційних моделей.

10. Назвіть класи, на які поділяють спеціалізовані мови імітаційного моделювання. Наведіть приклади мов моделювання серед кожного класу

спеціалізованих мов. З'ясуйте, за якими ознаками розрізняють мови імітаційного моделювання.

11. Охарактеризуйте основні етапи побудови імітаційної моделі та наведіть відомі вам приклади їх формулювання.

12. У чому полягає суть постановки задачі імітаційного моделювання?

13. Охарактеризуйте етап визначення задачі та її аналіз.

14. Опишіть етап збирання інформації.

15. Опишіть етап висування гіпотез і прийняття припущень.

16. Опишіть етап встановлення основного змісту моделі. З яких компонентів складається реальна обстановка задачі? Наведіть приклади.

17. Опишіть етап визначення параметрів, змінних і критеріїв ефективності.

18. Опишіть суть концептуальної моделі задачі «Визначення оптимального правила пріоритету в календарному плануванні засобами імітаційного моделювання».

19. Поясніть логічну структурну схему імітаційної моделі завантаження ЕОМ, користуючись рекомендованою літературою

20. Розгляньте історію зародження методу Монте-Карло. Наведіть приклади застосування методу Монте-Карло у різних галузях науки і техніки. Розкажіть про зв'язок методу Монте-Карло із задачами теорії ймовірностей, математичної статистики та обчислювальної математики.

21. Наведіть приклади, коли доцільно в методі Монте-Карло застосовувати безпосередньо емпіричні дані та коли є рація скористатися одним з апроксимуючих теоретичних розподілів. Відмітьте переваги та недоліки використання емпіричних даних. Подумайте, чому іноді дуже важливо застосовувати теоретичні розподіли.

22. Поясніть розв'язання задачі обчислення визначеного інтеграла методом Монте-Карло. Чому при обчисленні багатократних інтегралів точність оцінки інтеграла не залежить від кратності інтеграла? Порівняйте цю властивість з відповідними характеристиками інших числових методів обчислення інтегралів.

23. Поясніть методику оцінки точності обчислення значення визначеного інтеграла за допомогою методу Монте-Карло. На прикладі розгляньте способи визначення кількості спроб в методі Монте-Карло залежно від заданої точності кінцевого результату. Що таке правило зупинки?

24. Охарактеризуйте рівномірно розподілену на відріжку $[0,1]$ випадкову величину. У чому полягає унікальна властивість розподілу ймовірностей даної величини і як вона використовується? Що таке РВП $[0,1]$?

25. Наведіть доказ принципової можливості отримання рівномірної випадкової послідовності чисел за допомогою ЕОМ. Чому випадкові числа, отримані за допомогою ЕОМ, називаються квазірівномірними? Від чого залежить

кількість випадкових чисел, що отримуються за допомогою ЕОМ. Яку максимальну кількість різних випадкових чисел можна отримати за допомогою ЕОМ?

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Замятіна О. М. Комп'ютерне моделювання. URL: <https://studizba.com/hs/156-mgu-im-lomonosova/files/1865-sem/2102-imitacionnoe-modelirovanie-v-issledovaniyah-i/9498-/218850-o.m.-zamjatina-kompjuternoe.html> (дата звернення: 05.06.2024).
2. Неруш В. Б., Курдеча В. В. Імітаційне моделювання систем та процесів. URL: <https://ela.kpi.ua/items/29886d33-e657-453c-a4a8-9ebff59301a3> (дата звернення: 05.06.2024).
3. Elberg M., Tsygankov N. Simulation modeling. URL: <http://simulation.su/uploads/files/default/2017-uch-posob-elberg-cigankov.pdf> (дата звернення: 05.06.2024).
4. Kelton V., Lowe A. Simulation modeling. Classic CS. URL: https://heterarchica.wordpress.com/wp-content/uploads/2017/01/1kel_ton_v_lou_a_imitacionnoe_modelirovanie.pdf (дата звернення: 05.06.2024).
5. Стеценко, І. В. Моделювання систем. URL: <https://duikt.edu.ua/ua/lib/1/category/1082/view/1740?lang=ua&act=view&page=1&category=1082&id=1740> (дата звернення: 05.06.2024).
6. Метод Монте-Карло суть та приклади застосування. URL: <https://intellect.icu/metod-monte-karlo-sushhnost-i-primery-primeneniya-2928> (дата звернення: 05.06.2024).
7. Метод Монте Карло – що це в економіці, приклади. URL: <https://aws.amazon.com/ru/what-is/monte-carlo-simulation/> (дата звернення: 05.06.2024).
8. Купалова Г. І. Теорія економічного аналізу. URL: <https://westudents.com.ua/knigi/188-teorya-ekonomchnogo-analzu-kupalova-g.html> (дата звернення: 05.06.2024).
9. Литвинов А. Л. Теорія систем масового обслуговування. URL: <https://eprints.kname.edu.ua/50287> (дата звернення: 05.06.2024).

МЕТОДОЛОГІЯ І ПРАКТИКА НАУКОВИХ ДОСЛІДЖЕНЬ ТА ІНТЕЛЕКТУАЛЬНА ВЛАСНІСТЬ

(укладач Пальчевський Б. О.)

ТЕМА 1 ПОНЯТТЯ ПРО НАУКУ ЯК СИСТЕМУ ЗНАНЬ

Наука – це система знань про природу, суспільство та мислення, яка створена в результаті їх вивчення. *Наукове дослідження* – це процес створення нових знань шляхом вивчення предмета чи явища з метою розкриття законів його виникнення, розвитку і перетворення в інтересах суспільства.

Наука виникла в результаті суспільного розподілу праці, при відділенні розумової праці від фізичної. До виникнення науки люди також мали певні знання про навколишнє середовище, однак ці знання були уривчасті та невпорядковані. *Наукові знання* представляють собою чітку систему *понять*, що відображають предмети природи та суспільства в їх загальних та суттєвих ознаках. Наприклад, явища електризації тіл при терті та іскріння наелектризованих тіл були відомі віддавна, однак тільки відкриття електрона дозволило об'єднати ці явища і створити науку про електрику. При створенні *понять* в процесі пізнання виділяється головне і суттєве, залишаючи без уваги несуттєві та другорядні ознаки, що дозволяє правильно і глибоко зрозуміти об'єкт вивчення. Для впорядкування розрізнених знань в систему, що представляє собою наукову *теорію*, використовуються загальнонаукові або філософські методи. Наприклад, відкриття законів інерції і падіння тіл, а також опис цих законів за допомогою введених методом абстрагування *понять* «маси», «інерції», «енергії», «прискорення» тощо створена механіка як наука. Метод абстрагування дозволив створити навіть такі наукові *поняття*, що не зустрічаються в матеріальному світі: «ідеальний газ», «крапка», «площина», «абсолютно тверде тіло» тощо, які однак відіграють важливу роль при створенні наукових теорій.

Практика є критерієм перевірки істинності теорії. За загальним, філософським розумінням практика є виробнича діяльність, спрямована на перетворення предметів та явищ таким чином, щоб їх форма та зміст вдовольняли потреби людини. Розвиток машинного виробництва сприяв швидкому перетворенню дослідницької діяльності в активний фактор суспільного виробництва. Воно вдосконалює, поповнює, систематизує та перевіряє на практиці систему наукових знань. В умовах сучасного науково-технічного прогресу наука вже не тільки супроводжує розвиток техніки, а випереджує її, направляючи розвиток матеріального виробництва на найбільш перспективні напрями.

Схематично процес наукового дослідження показано на рисунку 1.1.



Рисунок 1.1 – Наукове дослідження: 1 – накопичення дослідних даних; 2 – побудова теорії; 3, 4 – експериментальна перевірка теорії та її практичне застосування

На схемі виділено два рівні наукових досліджень:

1) *емпіричний*, в основі якого є накопичення нових фактів, їх аналіз, узагальнення з метою отримання залежностей, придатних для практичного використання;

2) *теоретичний*, на якому здійснюється узагальнення та створюється система знань, що описують закономірності певної області знань. На цьому етапі створення системи знань проходить через ряд етапів, а саме:

наукова ідея – гіпотеза – теорія.

Наукова ідея пояснює нове явище без обґрунтування зв'язків, що використанні для інтуїтивного пояснення явища. Вона витікає із вже накопичених раніше знань, але при цьому дозволяє відкрити нову, ще невідому закономірність.

Гіпотеза є результатом матеріалізації наукової ідеї, коли передбачається причина, що викликає нове досліджуване явище. Г. Ом, вивчаючи явище опору провідників, звернувся до роботи Д.Фуре «Аналітична теорія теплоти». Він висунув ідею, що явище проходження електричного струму по провіднику є аналогічним до явища поширення потоку тепла. На основі цієї ідеї він висуває гіпотезу, що величина струму залежить від прикладеної різниці потенціалів, по аналогії до різниці температур, що викликає потік тепла.

Якщо гіпотеза узгоджується із результатами досліджень, то її називають теорією або законом. При проведенні наукового дослідження кожна гіпотеза експериментально перевіряється. При подальшому накопиченні фактів одна гіпотеза може уточнюватися або замінюватися новою, яка краще пояснює як старі, так і нові дані. Результатом логічного розвитку та вдосконалення гіпотези є її перетворення в закон.

Закон – це внутрішній, суттєвий зв'язок явищ, який обумовлює їх закономірний розвиток. Він відображає стійкий зв'язок між явищами та властивостями матеріальних об'єктів.

При проведенні наукових досліджень обидва рівні використовуються однаково успішно, доповнюючи один одного. Результати емпіричного рівня дослідження слугують вихідним матеріалом для створення теорії, перевірки її істинності і наступного розвитку та вдосконалення. З іншого боку, теорія дозволяє виділити суттєві зв'язки при проведенні емпіричних досліджень, підвищити точність його результатів, пояснити та узагальнити їх, вказати найбільш перспективні області дослідження, як це стало, наприклад, в хімії після відкриття періодичної системи елементів Д. І. Менделєєва. Теорія уточнює наукові знання за допомогою більш містких понять, впорядковує їх, створюючи умови для практичного використання. В середньовіччі, наприклад, арифметика викладалась в університеті, бо маніпуляції із використовуваними в той час римськими цифрами були складними. Введення десяткової системи числення та арабських цифр значно спростило всі математичні дії.

Створення та розвиток системи наукових знань здійснюється шляхом пізнання оточуючого нас матеріального світу – його предметів, явищ, їх ознак і відношень. Цей процес пізнання ототожнюється із науковим дослідженням. Відправним пунктом пізнання є чуттєве споглядання предметів та явищ, їх відображень в свідомості людини в вигляді чуттєвих образів. Ці образи відтворюють зовнішню сторону об'єкту пізнання. Завданням наукового пізнання є знаходження закономірностей матеріального світу, тобто закономірностей, що відтворюють загальне, суттєве в об'єкті. А для цього необхідно застосувати абстрактне мислення, формами якого є поняття, судження та умовивід.

Поняття – дефініція, що відображає загальні та суттєві ознаки предмету. З його допомогою людина проникає в суть предметів та явищ матеріального світу. Воно є основною цеглинкою, з допомогою яких будується споруда системи наукових знань. Широковідомими є поняття «маси», «енергії», «в'язкості», «ньютонівської рідини», «пружного тіла Гука» тощо. Поняття можуть бути загальними, одиничними та узагальнювальними, абстрактними та конкретними, абсолютними або відносними. Взаємозв'язок понять здійснюється за допомогою суджень.

Судження – така форма думки, в якій сполучають поняття, щось стверджуючи або заперечуючи про реальні предмети та явища. Наприклад, «мідь – метал», «всі метали – пластичні» тощо. Сполучаючи декілька понять, ми створюємо судження про предмети матеріального світу. Зв'язок суджень між собою здійснюється за допомогою умовиводу.

Умовивід (висновок) – форма мислення або логічна дія, в результаті якої із одного або декількох відомих нам і певним чином зв'язаних суджень виводиться нове знання про предмети та явища матеріального світу. Наприклад, умовиводом

буде така операція мислення: «всі метали пластичні» + «мідь – метал» = «мідь пластична». Сполучення двох відомих нам суджень дало можливість перейти від відомого до невідомого, тобто отримати нове знання.

Таким чином, результатом наукового дослідження є узагальнення невпорядкованих уявлень про закономірності природи, суспільства та мислення в вигляді законів науки, що описують внутрішній, суттєвий зв'язок явищ або ознак матеріального світу.

Розглянемо, які методи використовуються для створення системи наукових знань.

Метод – система правил та способів підходу до вивчення явищ і закономірностей природи, суспільства та мислення. Знання методу задає спосіб теоретичного дослідження або практичного здійснення чогось, який ґрунтується на знаннях закономірностей розвитку об'єктивної дійсності і досліджуваних предмету чи явища. Знання методу орієнтує дослідника, допомагає йому вибрати суттєве, намітити шлях від відомого до невідомого.

Розрізняють *всезагальний метод* (діалектика), *загальнонаукові або філософські методи*, які застосовуються у всіх науках та в практичній діяльності людей, а також *конкретно-наукові або спеціальні методи*, які застосовуються в межах однієї або декількох суміжних наук.

Розглянемо детальніше найбільш поширені загальнонаукові методи. Частина з них застосовується при теоретичному рівні дослідження, частина – на емпіричному. Є серед них і такі, що застосовується на обох рівнях наукового дослідження.

До найбільш поширених методів відносяться аналіз і синтез, індукція і дедукція, абстрагування та конкретизація.

Аналіз та синтез.

Під *аналізом* розуміють метод дослідження, який полягає в тому, що об'єкт дослідження подумки або практично розділяється на складові елементи. Це можуть бути частини об'єкта або його ознак, властивостей, відношень. Кожна із виділених складових частин досліджується окремо як частина цілого. *Синтез* дозволяє здійснити об'єднання частин об'єкту, встановити їх зв'язки та описати об'єкт як єдине ціле.

В науковому дослідженні аналіз і синтез взаємопов'язані, вони доповнюють один одного і можуть по чергово повторюватись декілька разів. Результати аналізу дозволяють вивчити конкретні факти, відкрити глибину явища. Одночасно, застосування синтезу дозволяє об'єднати окремі факти, створити теорію, побачити перспективи розвитку процесу, його зв'язок з іншими явищами.

Прикладом застосування взаємопов'язаних аналізу та синтезу може бути робота Дж. Уатта над паровою машиною. При аналізі він розділив її робочий цикл пароутворення на три частини: утворення, розширення та конденсація пари. Для реалізації кожної частини циклу він розробив окремі пристрої. Потім шляхом об'єднання цих пристроїв отримав парову машину, яка реалізувала при роботі всі три частини циклу.

Індукція та дедукція.

Наукова індукція – це такий умовивід, в якому загальний висновок про ознаки якоїсь множини елементів зроблений в результаті дослідження певної частини елементів цієї множини. *Дедукція* – умовивід, в якому висновок про якийсь елемент із множини робиться на основі знання загальних властивостей цієї множини.

Індукція і дедукція є взаємопротилежні методи дослідження, вони використовуються в науковому дослідженні у взаємозв'язку, доповнюючи один одного. Процес наукового пізнання здійснюється від індуктивного узагальнення до дедуктивного висновку, перевірці цього висновку і до більш глибокого узагальнення.

Найбільш поширена помилка при їх застосуванні – це поспішність індуктивного узагальнення, узагальнення без достатнього обґрунтування або за другорядними ознаками, підміна причинних зв'язків випадковою часовою послідовністю, тобто випадки необґрунтованого поширення отриманого висновку за межі тих конкретних умов, для яких він був отриманий. Наприклад, при вивченні трьохчлену Ейлера $y = x^2 + x + 41$ матимемо таку картину. При послідовних значеннях x : 0; 1; 2; 3; 4;...; 10 отримаємо значення величини y відповідно: 41; 43; 47; 53; 61; 71; 83; 97; 113; 131; 151; ..., тобто прості числа. Роблячи індуктивне узагальнення, будемо стверджувати, що при всіх цілих додатних x значення y представлятиме собою ціле число. Однак такий індуктивний висновок буде поспішним, оскільки перевірка його показує, що при значенні $x=40$ трьохчлен дорівнюватиме $y=41^2$, тобто складному числу.

Абстрагування та конкретизація.

Абстрагування – це метод наукового пізнання, що полягає в мисленому виділенні цікавих для дослідника ознак, зв'язків і відношень предмету чи явища та їх мислене відривання від всіх інших. Під час абстрагування здійснюється відкидання несуттєвих, другорядних ознак, зв'язків, відношень предмету чи явища, що утруднюють їх вивчення. Результатом абстрагування є створення абстракцій, найбільш поширеними з яких можна вважати наступні три види.

Абстракція ототожнення, що отримується шляхом відокремлення від індивідуальних ознак предметів та виділення узагальненої ознаки. Вона

застосовується для утворення поняття якогось класу предметів і містить ознаки, що відрізняють цей клас від інших («метали», «деталі», «машини» тощо).

Абстракція ізоляції утворюється шляхом мисленого відокремлення та фіксації певних ознак і відношень від предметів та явищ, з якими вони пов'язані. В цьому випадку утворюють загальні абстрактні поняття, наприклад: «точність», «теплопередача», «надійність» тощо.

Абстракція ідеалізації – це результат абстрагування, який дозволяє створити поняття, що не існують в реальному світі. В цьому випадку певні ознаки чи властивості об'єкту вивчення доводять до граничного значення, наприклад: «точка», «ідеальний газ», «абсолютно чорне тіло» тощо.

В процесі наукового пізнання метод абстрагування тісно пов'язаний з *конкретизацією*. Виявивши за допомогою абстрагування від конкретних фактів загальні закономірності розвитку предмету або явища, що вивчаються, дослідник знову повертається до конкретного. Розглянемо приклад застосування цих методів у взаємозв'язку.

Якщо на поверхню будь-якого тіла попадає світло, то частина світла відбивається, а решта поглинається цим тілом, віддаючи свою енергію. Чим більше світла поглинає тіло, тим темнішим воно виглядає. Абстрагуючись від конкретних фізичних тіл, дослідники ввели поняття «абсолютно чорне тіло», розуміючи під ним таке, яке поглинає все світло, що на нього попаде. Таких тіл в природі немає, однак легко визначається залежність температури «абсолютно чорного тіла» від світлового потоку. Отримана залежність конкретизується для опису реальних тіл, які поглинають тільки частину світлової енергії, шляхом введення коефіцієнта поглинання. Коефіцієнти поглинання визначаються експериментально для реальних фізичних тіл.

ТЕМА 2 МЕТОДОЛОГІЧНІ АСПЕКТИ НАУКОВОГО ДОСЛІДЖЕННЯ

Сучасна промисловість характеризується швидким розвитком технологій та технічних засобів виробництва. Технологічні машини стають все більш швидкодіючими та продуктивними, мають кращий коефіцієнт корисної дії та вищу надійність, а технології – більш ефективними та досконаліми.

Вдосконалення технологічних машин супроводжується зростанням їх складності (в середньому вдвічі кожні 15 років). При цьому виникає ряд протиріч, що стримують їх розвиток. Відмітимо два основних:

1) переважання темпів зростання складності технологічних машин над темпами розвитку методів їх проектування;

2) погіршення співвідношення між тривалістю їх розробки та часом морального старіння (в ХХ столітті продуктивність праці зросла в 10 раз, а продуктивність проектування – тільки на 20%).

Без уміння вивчити сучасну технологічну машину та умови її функціонування стає неможливим її вдосконалення, розробка та експлуатація. А таке вивчення представляє собою наукове дослідження.

Тісний зв'язок науки та виробництва визначив і нероздільність наукової та інженерної діяльності. Саме вміння провести наукове дослідження відтіняє творчість інженера, надає їй витонченості при прийнятті нестандартних рішень. Побудова математичної моделі є сполучною ланкою наукового дослідження з проектуванням та вдосконаленням технологічної системи. Наявність етапу математичного моделювання створює методичну спільність наукових досліджень та процесу проектування чи оптимізації технологічної системи. В результаті проведення цих етапів не просто виникає нове знання, але і здійснюється його конкретне застосування.

Наукова діяльність, як і будь-яка інша, здійснюється за допомогою певних засобів, а також особливих прийомів і способів, від правильного використання яких багато в чому залежить успіх. Сукупність цих прийомів позначають поняттям «метод». Метод з грецької означає шлях пізнання.

Метод наукового дослідження – це спосіб пізнання об'єктивної реальності, система правил і прийомів, за допомогою яких досягається об'єктивне пізнання дійсності. Сам спосіб представляє собою певну послідовність дій, прийомів, операцій.

Методика дослідження – це система правил використання методів, прийомів та способів для проведення дослідження. Отже, метод орієнтує у вирішенні конкретного завдання, досягнення результату в будь-якій сфері діяльності.

Методологія – це учення про науковий метод пізнання (теорія методів дослідження), про систему знань про теорію науки або систему методів дослідження.

У самому загальному сенсі під методологією розуміється система методів, використовуваних в деякій області діяльності.

Вирізняють методи (рис. 2.1):

- загально-філософські,
- загальнонаукові, які застосовують у дослідницькому процесі різних наук,
- конкретно-наукові, які використовують – при розв'язанні конкретних наукових задач.

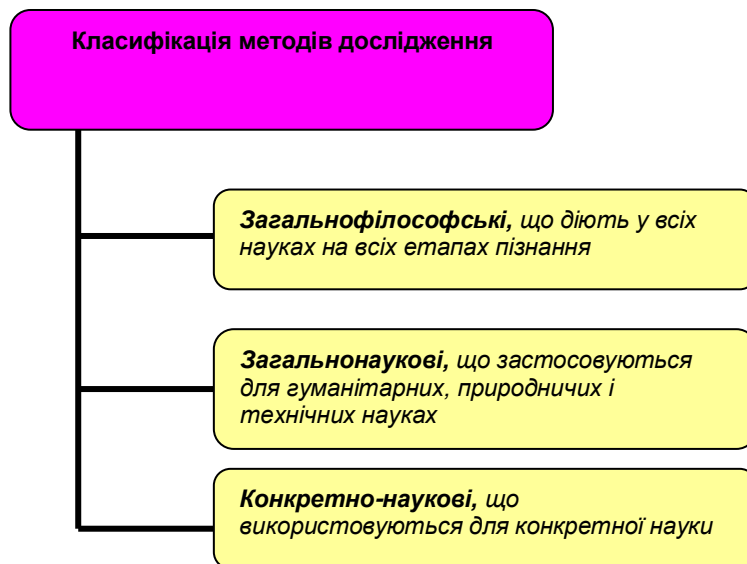


Рисунок 2.1 – Класифікація методів дослідження

Загально-філософські методи.

Фундаментальним, узагальненим методом пізнання дійсності є діалектичний метод. Діалектичний підхід дозволяє обґрунтувати причинно-наслідкові зв'язки, процеси диференціації та інтеграції, постійне протиріччя між сутністю і явищем, змістом і формою, об'єктивність в оцінці дійсності.

Поняття являється основою цеглиною, за допомогою якої створюється система наукового знання. Взаємозв'язок понять робиться шляхом суджень і висновків.

Судження – це така форма мислення, за допомогою якої поєднують поняття, що-небудь стверджуючи або заперечуючи про реальні предмети і явища. Зв'язок суджень здійснюється за допомогою висновків.

Умовивід (висновок) – розумова операція, за допомогою якої з певної кількості заданих суджень виводиться інше судження, яке певним чином пов'язано з вихідним. За допомогою висновку з одного або декількох відомих нам і певним чином пов'язаних суджень виводиться нове знання про предмети і явища реального світу. Наприклад:

- 1) усі метали пластичні;
- 2) мідь – метал;
- 3) мідь пластична.

Специфіка філософських методів полягає в тому, що вони задають лише найзагальніші правила дослідження, його генеральну стратегію.

Загальнонаукові методи дослідження.

Розвиток науки відбувається на основі широкого використання *загальнонаукових методів*. Це сукупність знань про принципи і методи, що застосовуються в будь-якій науковій дисципліні.

Загальнонаукові методи дослідження умовно поділяють на три групи:

1. загальнологічні методи, що використовуються як на теоретичному, так і на емпіричному рівнях дослідження (аналіз і синтез, індукція та дедукція, аналогія, моделювання тощо).

2. теоретичні методи – методи, що використовуються на теоретичному рівні дослідження. Особливість теоретичного дослідження полягає у використанні абстрактних уявлень, ідей, положень, концепцій.

3. емпіричні – методи емпіричного дослідження (спостереження, експеримент, вимірювання, порівняння).

Розглянемо окремо кожен групу загальнонаукових методів.

Методи, що використовуються на теоретичному і емпіричному рівнях дослідження.

До них відносяться аналіз і синтез, індукція і дедукція, аналогія і моделювання, абстрагування і конкретизація, класифікація (рис. 2.2).

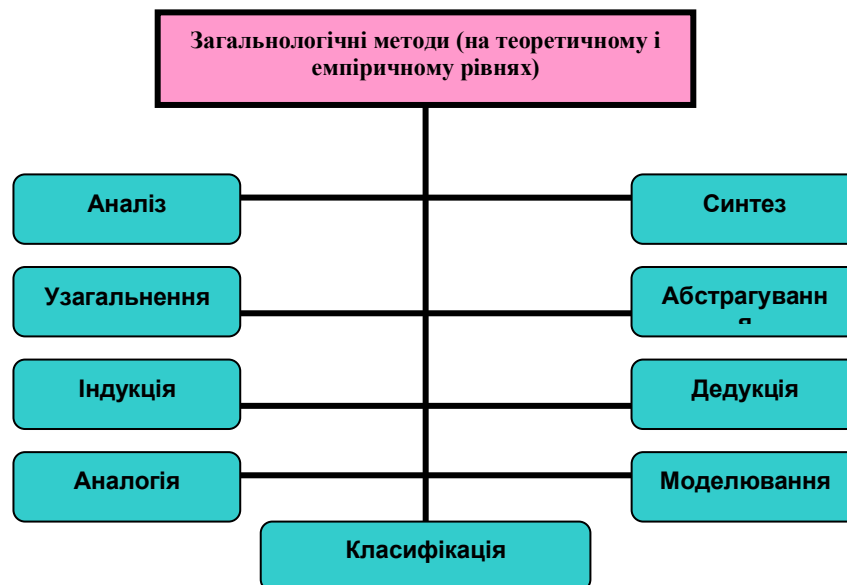


Рисунок 2.1 – Класифікація загальнологічних методів дослідження

Аналіз – це метод дослідження, що полягає в тому, що предмет вивчення подумки або практично розчленовується на складові елементи (частини об'єкту або його ознаки, властивості, відносини.). Кожна з виділених складових частин досліджується окремо як частини цілого.

Синтез – об'єднання пізнаних в результаті аналізу елементів в єдине ціле. Синтез дозволяє здійснити з'єднання частин предмета, розчленованого в процесі аналізу, встановити їх зв'язок і пізнати предмет як єдине ціле.

Аналіз і синтез взаємозв'язані. Приступаючи до аналізу, дослідник вже має якесь загальне уявлення про об'єкт вивчення як ціле. Таким чином, ми бачимо, що аналіз вже з самого початку дослідження здійснюється спільно з синтезом.

Дж. Уатт, наприклад, розділив робочий цикл парової машини на три частини: утворення, розширення і конденсація пари. Для реалізації кожної частини циклу були розроблені відповідні пристрої, виходячи з необхідності забезпечити роботу парової машини як цілого.

Узагальнення – процес уявного переходу від одиничного до загального, від менш загального, до більш загального.

Абстрагування (ідеалізація) – це метод наукового пізнання, в процесі якого відбувається відкидання несуттєвих, побічних ознак, зв'язків і відносин предмета або явища, що утрудняють проведення дослідження, тобто відділення істотного від випадкового. При цьому, як результат абстрагування, створюються абстракції.

Абстрагування в процесі наукового пізнання тісно пов'язане з *конкретизацією*.

Розглянемо приклад використання абстрагування і конкретизації в науковому пізнанні. Якщо на поверхню будь-якого тіла потрапляє світло, то частина його відбивається, а частина проникає всередину, віддаючи тілу свою енергію. Чим більше світла поглинає тіло, тим темніше воно здається. Абстрагуючись від конкретних фізичних тіл, учені створили поняття «Абсолютно чорне тіло», тобто тіло, яке поглинає все світло, що падає на нього. Таких тіл в природі немає, але при вивченні властивостей «абсолютно чорного тіла» легко визначити його залежність від температури. При вивченні реальних тіл конкретизація дозволяє врахувати те, що вони поглинають тільки частину енергії, що визначається коефіцієнтом поглинання. Введення цього коефіцієнта дає можливість застосувати закон, отриманий для ідеального об'єкту, для реальних тіл, визначивши експериментально їх коефіцієнти поглинання.

Індукція – процес виведення загального положення з спостереження низки часткових одиничних фактів, тобто пізнання від часткового до загального. В процесі наукового пізнання індукція завжди використовується в нерозривному зв'язку з дедукцією.

Дедукція – процес аналітичного переходу від загальних представлень до часткових.

ТЕМА 3 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ НАУКОВОГО ДОСЛІДЖЕННЯ

3.1 Пошук матеріалів за темою дослідження

Алфавітний каталог, в якому література розташовується за алфавітним порядком авторів або за алфавітним порядком публікацій (для словників, збірників, альбомів, інструкцій, звітів та для будь-яких публікацій, якщо авторів більше трьох).

Систематичний каталог, в якому література розташовується за універсальним десятковим класифікатором (УДК), за яким всі галузі знань розділені на 10 класів:

0 – Загальний відділ. Наука. Організація. Розумова діяльність. Знаки і символи. Документи і публікації.

1 – Філософія.

2 – Релігія.

3 – Економіка.

4 – Вільний клас.

5 – Математика. Природничі науки.

6 – Прикладні науки. Медицина. Техніка.

7 – Мистецтво. Прикладне мистецтво. Фотографія. Музика.

8 – Філологія. Мовознавство. Художня література. Літературознавство.

9 – Краєзнавство. Географія. Біографія. Історія.

Кожен із основних класів розділений на 10 розділів, які, в свою чергу, на десять підрозділів і так далі. Деталізація понять здійснюється шляхом подовження індексів. Кожна наступна цифра не змінює змісту попередніх, а тільки їх уточнює. Розглянемо, наприклад, розподіл на групи і підгрупи класу прикладних наук.

УДК 62. Інженерна справа. Техніка.

УДК 621. Машинобудування.

УДК 621.7 Обробка в цілому.

УДК 621.9 Різання зняттям стружки.

Для розширення можливостей системи УДК використовуються також спеціальні визначники.

* *Визначники характеру матеріалу*, вказуються в дужках: (031) – Енциклопедія; (063) – Матеріали з'їзду; (075) – Підручники. Наприклад, УДК 621.9 (075) – Підручники з різання матеріалів зняттям стружки.

* *Визначники аспекту вивчення питання*, вказуються після основного індексу через крапку і два нулі перед цифровим позначенням: .001 – Теоретична точка зору; .002 – Технологія; .002.3 – Сировина. Матеріали.

Застосування двокрапки дозволяє здійснювати синтез понять. Наприклад, поняття УДК 621.9 : 536.2 значить «теплопередача при різанні зняттям стружки», бо двокрапка об'єднує поняття УДК 621.9 «Різання зняттям стружки» та УДК 536.2 «Теплопередача».

Предметно-алфавітний каталог використовується, коли невідомо, до якого розділу УДК відноситься тема дослідження. Він побудований за допомогою набору ключових понять, які розташовані в ньому в алфавітному порядку. При користуванні цим каталогом ключове поняття треба висловити максимальною кількістю слів-синонімів. Якщо в каталозі такого слова немає, то його замінюють більш загальним поняттям.

3.2 Пошук патентних матеріалів за темою дослідження

Пошук патентних матеріалів за темою дослідження також є невід'ємною частиною етапу постановки задачі дослідження, оскільки він забезпечує:

- прогнозування тенденції розвитку наукових напрямків розвитку техніки та технологій;
- оцінку технічного рівня результатів шляхом їх порівняння з існуючими зразками;
- перевірку патентоспроможності створених технічних систем.

Патентна документація – це найбільш повне зібрання даних про науково-технічні досягнення людства за останні 250 років. Основою класифікації цієї інформації є міжнародний класифікатор винаходів (МКВ), побудова якого ґрунтується на двох підходах:

- *предметно-тематичний*, за яким об'єкти класифікуються в залежності від галузі їх застосування;
- *функціональний*, за яким в основу класифікації покладено тотожність функцій об'єктів незалежно від галузі техніки, де об'єкт застосовується.

Система МКВ включає 8 розділів, 20 підрозділів, 115 класів і так далі. Основні її розділи є:

- A – Задоволення життєвих потреб людини;
- B – Різні технологічні процеси;
- C – Хімія і металургія;
- D – Текстиль та папір;
- E – Будівництво;
- F – Прикладна механіка, освітлення і опалення; двигуни та помпи; зброя та боєприпаси;
- G – Технічна фізика;
- H – Електрика.

За предметно-тематичною ознакою побудовані розділи А, С, D, E, F, G, H, а за функціональною – розділ В.

Особливу цікавість для дослідника на етапі постановки задачі дослідження та вибору напрямку проведення робіт представляє аналіз патентів по темі дослідження за останній 5-10 років. Підраховуючи тільки їх кількість, вже можна зробити наступні висновки.

Якщо кількість патентів за кожен наступний рік перевищує дані попереднього року, то напрям дослідження є перспективним, а його тема – актуальною.

Якщо кількість патентів є приблизно однаковою, то необхідно створити паралельний напрям проведення дослідження, а саму наукову тему розширити.

Якщо кількість патентів з кожним роком зменшується, то тема дослідження ґрунтується на застарілих ідеях і представленнях. В цьому випадку буде доцільним проведення пошукових досліджень для знаходження нових принципів.

ТЕМА 4 ВИБІР НАПРЯМКУ ТЕМИ ДОСЛІДЖЕННЯ

Визначення напрямку дослідження. На сучасному етапі розвитку науково-технічного прогресу технічні системи розвиваються безперервно, характеризуються високим ступенем спадковості. Кожна заново створена технічна система є результатом попередніх наукових праць, процесів, пристроїв, речовин. В механіці, фізиці, хімії, в технічних науках джерелом гіпотез для побудови нових теорій, принципів завжди були старі. І. Ньютон писав: «Якщо я бачив далі своїх попередників, то тільки тому, що я стояв на плечах гігантів».

Тому, починаючи дослідження, необхідно оглянути матеріали по даній темі та провести їх аналіз з точки зору поставленої мети наукової роботи. До таких матеріалів може відноситися науково-технічна література (книги, науково-технічні збірники, реферативні та інформаційні видання тощо) та патентна література (реферативні журнали винаходів та описи винаходів).

Вивчення стану проблеми є необхідним етапом наукового дослідження, бо більшість задач дослідження виникає безпосередньо на виробництві. Сутність такої задачі, як правило, завуальована багатьма несуттєвими і другорядними фактами. Тому для правильного формулювання задачі дослідження необхідно провести певну роботу.

Наприклад, коли на Обухівському заводі в Санкт-Петербурзі перейшли до випуску тяжких гармат, перед дирекцією постала проблема забезпечення їх якості: нові гармати, виготовлені із 12-тонних заготовок, то розвалювались після декількох пострілів, то витримували декілька тисяч пострілів. Було вирішено

перевірити витримування технологічних вимог на всіх етапах виготовлення гармат. Для цього технологічний процес розділили на ряд послідовних етапів. Інженерові-технологіві Д. К. Чернову необхідно було «забезпечити кування гарматних заготовок в однакових умовах відповідно до письмової вказівки власника заводу». Д. К. Чернов, співставляючи зразки металу міцних та неміцних гармат, визначив, що у міцних гармат метал дрібнозернистий, а в неміцних – великозернистий. Уточнюючи мету дослідження, він припустив, що міцність металу визначається його зернистістю. Аналіз науково-технічної літератури дозволив Д. К. Чернову визначити, що структура металу формується при куванні. Тепер задача дослідження сформулювалась так: «вивчити вплив режимів кування на структуру та зернистість металу, а також визначити режими, при яких структура забезпечує високу міцність металу». Експериментальні дослідження, проведені на Обухівському заводі, показали, яким чином режими теплової і механічної обробки при куванні впливають на структуру і міцність металу.

Із наведеного прикладу бачимо важливість правильної постановки задачі дослідження. Кожне нове дослідження, наукова публікація, що його відображає, виникають на ґрунті попередніх. Робота з науково-технічною літературою дозволяє визначити напрямок розвитку технічної системи (об'єкту чи процесу), відділити задачу від другорядних та примикаючих до неї, провести декомпозицію мети дослідження і провести його планування.

Конкретизація умов наукового дослідження. На кожен реальний об'єкт дослідження, технічну систему діють всі закони природи – не тільки відомі, але і ще невідкриті. Тому при проведенні наукового дослідження ми змушені спрощено розглядати предмети та явища, що вивчаються, приймаючи до уваги тільки їх головні, суттєві сторони. Для відсіву факторів, вплив яких є несуттєвим, приймаються відповідні допущення. Багато допущень приймається несвідомо, що часто призводить до необґрунтованого розширення застосування результатів дослідження.

Розглянемо, наприклад, що необхідно було би врахувати, щоб врахувати абсолютно точно силу, з якою в даний момент яблуко тисне на стіл. Загальновідоме розв'язання, за яким сила F , з якою яблуко діє на стіл, дорівнює його вазі яблука P не враховує багатьох факторів. Проаналізуємо залежність $F = P = m \cdot g$, де m – маса яблука, g – прискорення сили земного тяжіння. Ці фактори непостійні в часі: маса яблука коливається, бо під дією тепла здійснюється випаровування води, фотосинтез тощо. Прискорення сили тяжіння залежить від географічної широти та висоти стола над рівнем моря, від переміщення мас всередині Землі тощо. Окрім того, на яблуко діє сила, що створена витісненням об'ємом повітря та багато іншого.

Цей приклад показує складність дослідження реального об'єкту, яка приводить до того, що кожне дослідження є наближеним. В залежності від потрібної вірогідності результатів, враховується більша чи менша кількість факторів.

Оскільки математична модель, отримана в результаті дослідження, не визначається однозначно вихідними умовами задачі, завжди буде корисно не вірити сліпо готовим моделям, а побудувати модель досліджуваного явища самостійно. В цьому випадку підкресляться прийняті допущення, виявляться неусвідомлено прийняті допущення. Отримана таким чином модель може бути порівняна з існуючими і, якщо наукові результати виявляться близькими, то таке дослідження може вважатися об'єктивним.

ТЕМА 5 СТРУКТУРА ПРИКЛАДНОГО НАУКОВОГО ДОСЛІДЖЕННЯ

Очевидно, що чим невизначеннішими є вихідні дані, тим складнішим буде наукове дослідження. В залежності від його мети – пізнавальної чи практичної – розрізняють *фундаментальні* та *прикладні* наукові дослідження. Якщо *фундаментальні дослідження* є необхідними для побудови системи наукових знань, її розвитку та вдосконалення і вирішують основну задачу – *що саме* може бути застосовано в практичній діяльності людини, то *прикладні дослідження* дають відповідь на інше питання – *як саме* можуть бути застосовані вже відомі наукові знання.

Фундаментальні дослідження направлені на відкриття і дослідження нових явищ та законів природи. Вони розширюють та вдосконалюють систему наукових знань про навколишній матеріальний світ. Результати наукових досліджень використовуються в практичній діяльності для створення нових джерел енергії, матеріалів, технологій. Ці дослідження проводяться на межі відомого та невідомого, тому вони мають найвищий ступінь невизначенності. При таких дослідженнях вибір напряму проведення дослідження, вибір самого об'єкту дослідження визначаються методом «спроб та помилок», а зменшення кількості дослідів та термінів проведення дослідження визначаються досвідом та кваліфікацією самого дослідника. Відомого багато прикладів із історії розвитку науки, коли вирішальну роль в науковому відкритті відігравав випадок.

Прикладні дослідження мають за мету знаходження шляхів застосування відкритих законів та явищ природи в практичній діяльності людини для створення нових машин, технологій та речовин. Кожне наукове відкриття повинно пройти довгий шлях, перш ніж воно втілиться в технічний об'єкт. Властивість провідника

нагріватися при проходженні електричного струму, наприклад, тільки згодом знайшла застосування в електронагрівальних приладах.

При проведенні прикладних наукових досліджень на основі наукових понять створюються цілеспрямовані технічні поняття. Наприклад, такі поняття технології металів як твердість, жаростійкість, холодоломкість тощо, що створені на основі понять фізики, хімії, виникли, виходячи із мети прикладної науки – отримання металу із заданими властивостями. Ступінь завершеності прикладного дослідження визначається придатністю його результатів для практичного застосування.

При *пошукових дослідженнях* метою є знаходження принципово нових шляхів створення технічних систем. Це може бути нове застосування відомого закону чи явища, використання нових властивостей матеріалів.

Науково-дослідна робота забезпечує створення нових технологій, машин, приладів та матеріалів, проведення оптимізації технічних систем з метою покращення їх експлуатаційних характеристик тощо.

Дослідно-конструкторська робота полягає в створенні певної конструкції технічної системи, що забезпечує виконання ним потрібної функції та задовольняє певні експлуатаційні обмеження.

Етапи прикладного дослідження.

Як здійснюється процес прикладного наукового дослідження? Чим прикладне дослідження принципово відрізняється від фундаментального? Що потрібно дослідникові, щоб вдало сформулювати реальну мету, досягнення якої повинна забезпечити технічна система? Щоб дати відповіді на ці питання розглянемо приклади простих наукових досліджень, в яких легко прослідковується методика їх виконання.

Розглянемо *структуру наукового дослідження*, проведеного Галілео Галілеєм для вивчення вільного падіння фізичних тіл. В той час вважалось загально визнаним положення Арістотеля, який стверджував, що важчі тіла падають швидше, ніж легкі, бо швидкість падіння є функцією маси тіла. Галілей провів аналіз твердження Арістотеля, порівнявши умови падіння важкого каменю та легкого листка. За Арістотелем, камінь повинен падати швидше від листка, бо він тяжчий, тобто швидкість каменю V_K більше швидкості листка V_L . Подумки з'єднаймо камінь та листок. Таке об'єднання повинно мати, з однієї сторони, проміжну швидкість V_{K+L} його елементів, тобто: $V_L < V_{K+L} < V_K$. Але, з іншої сторони, загальна маса такого об'єднання перевищить масу каменю, тому швидкість каменю і листка, за твердженням Арістотеля, повинна перевищити швидкість каменю. Для розв'язання цього протиріччя Галілей висунув гіпотезу, що швидкість падіння не є функцією маси тіла, що падає. Ця гіпотеза представляє

собою точне формулювання задачі дослідження, яке дозволило визначити фактори, що можуть впливати на експеримент, та фактори, вплив яких можна вважати несуттєвим. Г.Галілей враховував при експерименті масу та форму тіла, але не враховував опір повітря, температуру повітря тощо. Гіпотеза знайшла своє підтвердження під час славнозвісного експерименту на Пізанській вежі, з якої було скинуто ядро масою 80 кг та мушкетну кулю масою 200 г. Обидва тіла кульової форми торкнулись поверхні Землі одночасно.

Наукове дослідження, проведене Г.Галілеєм, має наступні етапи, які є характерними і для сучасного наукового дослідження:

- 1) *постановка задачі дослідження* як результат вивчення стану проблеми;
- 2) *виділення досліджуваного об'єкта із оточуючого середовища*, визначення його зв'язків, діючих обмежень, параметрів які впливають на результат дослідження;
- 3) *експериментальне дослідження та опис явища* вільного падіння.

Проведемо аналіз *структури наукового дослідження*, якщо об'єктом виступає технічна система. Перш за все, відмітимо, його цілеспрямованість, яка визначається більш загальною метою, а саме – створенням технічної системи для виконання заданої службової функції. Значну роль тут відіграють результати аналізу та наукового дослідження, представлені в вигляді математичних моделей функціонування технічної системи, які використовуються для цілеспрямованого вибору кращого варіанту її конструкції чи оптимізації режимів функціонування. В цьому випадку скорочуються терміни проектування технічних систем, зменшується ризик прийняття невірних рішень.

Таким чином, можна вважати, що процес прикладного наукового дослідження повинен бути підпорядкований вимогам проектування та оптимізації технічної системи. Він може бути розділений на чотири етапи.

1. *Постановка задачі дослідження як результат вивчення стану проблеми та мети інженерної творчої діяльності.*
2. *Визначення зв'язків досліджуваного об'єкта із оточуючим середовищем, діючих обмежень, параметрів, які впливають на результат дослідження.*
3. *Експериментальне дослідження, побудова математичної моделі досліджуваного об'єкта, перевірка її придатності для практичного застосування.*
4. *Використання математичної моделі для проектування чи оптимізації технічної системи.*

ТЕМА 6 МЕТОДОЛОГІЯ ТЕОРЕТИЧНОГО ДОСЛІДЖЕННЯ

Застосування теоретичних методів в дослідженні.

При системному представленні об'єкту дослідження, знання елементів об'єкту не дає знань про сам об'єкт, оскільки останній представляє собою не тільки суму частин, але і їх функціональний взаємозв'язок. Тому *методика застосування системного підходу до теоретичного аналізу* полягає в знаходженні в складному об'єкті простих елементів та зв'язків між ними, їх вивчення та опис за допомогою математичних моделей, а також їх з'єднання в модель складного об'єкта. Це агрегування моделей.

Такий підхід до аналізу складних задач є загальновідомим. Ще в середньовіччі застосовували декомпозицію складних задач на окремі простіші елементи, вивчали кожну з них. Рене Декарт в роботі «Розмірковування над методом» пише: «Розділіть кожну задачу на стільки частин, на скільки зможете і на скільки це потрібно вам, щоби її було легко розв'язати». Таке розділення дозволяє спростити об'єкт аналізу.

Однак, математик Г.Лейбніц, автор комбінаторики, відмітив: «Це правило Декарта малоефективне, оскільки мистецтво розділення залишається поза тлумаченням... Розділивши задачу на невідповідні частини, недосвідчений дослідник може тільки збільшити свої труднощі».

Питання декомпозиції.

Оскільки кожен складний об'єкт дослідження може бути розділений різними способами, то ця обставина стає вирішальною при проведенні теоретичного аналізу. Кількість виділених різнотипних елементів повинна бути найменшою, самі ці елементи повинні легко формалізуватися і описуватися простими математичними моделями. Тоді виділені елементи представлятимуть собою цеглини, з яких побудується модель складного об'єкту.

Так, наприклад, відкриття електрона, як основного елементу опису електричних явищ, дозволило створити теорію електрики, яка пояснювала з єдиних позицій такі явища як електризація тіл при терті, блискавка, електричний струм в металах, електроліз тощо. При цьому знання частин; елементів об'єкту ще не є знанням самого об'єкту, оскільки об'єкт – це не проста сума частин, а їх взаємозв'язок.

Таким чином, при проведенні теоретичного аналізу об'єкт розділюється на кінцеве число підсистем, а ті, в свою чергу, діляться на дрібніші аж до моменту, коли стануть достатньо простими для вивчення та математичного опису. Вони і будуть *елементами* опису складного об'єкту. Оскільки розділення складного об'єкту здійснюється із збереженням зв'язків між виділеними елементами, то

математична модель складного об'єкту складатиметься із математичних моделей елементів та математичних моделей зв'язків між ними.

Розглянемо на простому та наглядному прикладі, який відображає методичні проблеми, що виникають при розчленуванні об'єкту для його математичного опису.

Розглянемо різні варіанти розділення та опису об'єкту, що наведений на рисунку 6.1.

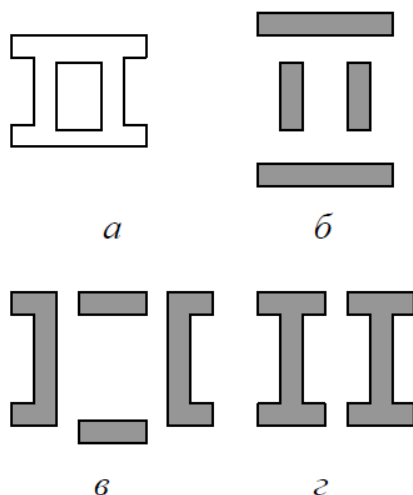


Рисунок 6.1 – Способи розчленування та опису об'єкту

Для кожного способу розділення його опис матиме вигляд:

- два паралельних бруски, що розділені двома короткими поперечинами, що трохи відстають від країв;
- два дзеркально симетричних елементи, що мають профіль швелера, які зверху та знизу розділені двома поперечинами;
- два двотаврових дзеркально симетричних елементи.

Із цього простого прикладу можна зробити такі висновки:

- 1) залежно від правильності розчленування процес вивчення об'єкта дослідження спрощується або ускладнюється;
- 2) виділення при розділенні більш складних елементів для опису і моделювання об'єкта дослідження спрощує опис зв'язків між ними і навпаки;
- 3) описаний об'єкт може використовуватись як елемент для опису і моделювання більш складних систем.

Таким чином, теоретичне дослідження – це знаходження в складному простих елементів і видів зв'язку між ними, вивчення і опис елементів, їх з'єднання в модель складного об'єкту. Кожний об'єкт можна розчленувати різними способами: ця обставина є вирішальною при проведенні теоретичних досліджень.

При аналізі явищ і процесів виникає потреба виділити головне. У цьому випадку може бути застосований метод *абстрагування*, тобто відхід від другорядних фактів з метою зосередитися на найважливіших.

У ряді випадків використовують спосіб *формалізації*, який полягає в тому, що основні положення процесів і явищ представляють у вигляді формул та спеціальної символіки, які дозволяють встановити закономірності між фактами, що вивчаються.

Математичне моделювання в теоретичних дослідженнях.

Процес моделювання – це вивчення властивостей об'єкта моделювання шляхом аналізу властивостей його моделі.

Модель – це об'єкт або опис об'єкта, системи для заміщення однієї системи (оригіналу) іншою системою для кращого вивчення оригіналу або відтворення будь-яких його властивостей. При моделюванні досліджується не сам об'єкт, а проміжний – допоміжний об'єкт, що здатний на окремих етапах пізнання представляти об'єкт пізнання, а також давати інформацію про нього.

Моделлю може бути об'єкт будь-якої природи, який замінює із будь-яким ступенем точності досліджуваній об'єкт. Досягнення цілей теоретичного дослідження без побудови моделі досліджуваного об'єкту практично неможливе. Кожна проектна та дослідницька діяльність так чи інакше пов'язана з побудовою моделі. Креслення технічної системи – машини чи вузла, схема системи керування також є моделями.

ТЕМА 7 ПРОВЕДЕННЯ ЕКСПЕРИМЕНТАЛЬНИХ ДОСЛІДЖЕНЬ

Найважливішою складовою частиною наукових досліджень є *експеримент*. Знання, отримане за допомогою спостереження, експерименту і соціально-виробничої практики людей, називається *емпіричним*.

Слово «експеримент» походить від латинського слова “*experimentum*” – «спроба», «досвід». У науковій мові та дослідницькій роботі цей термін звичайно використовується у значенні, загальному для цілої низки тотожних понять: досвід, цілеспрямоване спостереження; відтворення об'єкта пізнання; організація особливих умов його існування; перевірка передбачення. В це поняття вкладається наукова постановка дослідів і спостереження явища, що вивчається, в умовах, які чітко контролюються і дозволяють слідкувати за його перебігом і відтворювати його кожного разу при повторенні цих умов. Саме по собі поняття «експеримент» означає дію, націлену на створення умов із метою здійснення того або іншого явища і, якщо це можливо, якомога чистішого, тобто, неускладненого іншими явищами.

Основною метою експерименту є виявлення властивостей досліджуваних об'єктів, перевірка справедливості гіпотез і, на цій основі, широке і глибоке вивчення теми наукового дослідження.

Для проведення експерименту будь-якого типу необхідно:

- розробити гіпотезу, що підлягає перевірці;
- створити програми експериментальних робіт;
- визначити способи та прийоми втручання в об'єкт дослідження;
- забезпечити умови для здійснення процедури експериментальних робіт;
- розробити шляхи та прийоми фіксування ходу і результатів експерименту;
- підготувати засоби експерименту (прилади, установки, моделі тощо);
- забезпечити експеримент необхідним обслуговуючим персоналом.

7.1 Види експериментальних досліджень технологічної системи

Реальні технологічні системи, що функціонують в складних умовах діючого виробництва, мають безмежну кількість властивостей. Все це затрудняє опис і дослідження таких систем за допомогою моделей. Оскільки в технологічній системі поєднані технічні засоби, предмет виробництва та виконавець, результатом взаємодії яких є реалізація технологічного процесу чи операції, то об'єктом дослідження може бути:

- вироби на різних стадіях виготовлення;
- технологічний процес та його операції;
- технологічний комплекс, оснащення, інструмент і допоміжні матеріали, що використовуються при виготовленні продукції;
- діяльність персоналу, що здійснює технологічний процес.

В практичній діяльності інженера-технолога експериментальні дослідження широко використовуються для створення нових технологічних процесів, для оптимізації та інтенсифікації технологічних систем, виявлення невикористаних резервів, створення системи оптимального керування та діагностування обладнання, або для покращення його якісних характеристик. В деяких виробництвах технологічні системи стали настільки складними, що не тільки їх проектування та оптимізація, але і нормальне функціонування неможливі без постійного супроводу експериментальними дослідженнями.

Виробнича система, як об'єкт експериментального дослідження, має певні особливості, а саме:

- наявність значної кількості різномірних елементів (матеріали, цілеспрямовані дії персоналу, технічні засоби, енергетичні впливи тощо);

- сильна дія випадкових факторів, що призводить до того, що, навіть при незмінному налагодженні виробничої системи, здійснюється розсіяння її вихідних характеристик.

Ці особливості визначають представлення виробничої системи чи її складових при експериментальному дослідженні як «чорну скриньку», коли не використовується інформація про структуру об'єкта. Тоді загальна схема опису виробничої системи визначиться знаходженням залежності вихідних параметрів Y від вхідних X , без врахування внутрішніх параметрів (рис. 7.1).

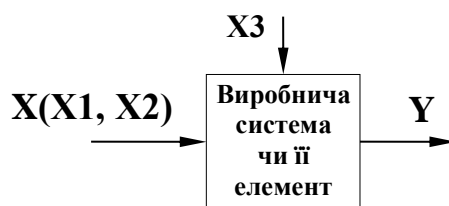


Рисунок 7.1 – Схема експериментального дослідження виробничої системи

Вектор вихідних параметрів Y відображає досліджувану характеристику виробничої системи (показник якості виробу, показник ефективності виробничої системи тощо).

Вектор вхідних керованих і контрольованих параметрів виробничої системи $X1$ включає як компоненти технологічні керовані і контрольовані параметри (швидкість різання, подача, глибина різання, температура спікання, концентрація фарби, тиск роботи розпилювача тощо) та конструктивні параметри (виліт різця, геометрія різальної частини інструменту, діаметр вихідного сопла розпилювача тощо).

Вектор вхідних некерованих але вимірюваних параметрів виробничої системи $X2$ включає такі обмежувальні параметри як припуск заготовки, міжцентрова відстань, найбільший діаметр сопла розпилювача тощо.

Вхідні збурюючі фактори $X3$ своєю дією впливають на протікання технологічного процесу, однак їх вплив не може бути ні проконтрольований, ні тим більш цілеспрямовано змінений (старіння обладнання, непередбачувані зміни властивостей сировини, дрейф режимів, зміна характеристик навколишнього середовища тощо).

Види виробничих експериментальних досліджень.

В залежності від того, який тип факторів враховується при експериментальному дослідженні (рис. 7.2), розрізнятимемо наступні види виробничих експериментальних досліджень:

- статистичний аналіз точності функціонування виробничої системи, $Y=f(X3)$, коли досліджується вплив випадкових збурень на вихідні показники (рис.,а), наприклад, статистичний аналіз якості продукції, аналіз стабільності протікання технологічного процесу тощо;
- пасивний експеримент, $Y=f(X2)$, який дозволяє визначити зв'язок між вхідними та вихідними параметрами в умовах дії збурень $X3$ при природному протіканні досліджуваних процесів;
- активний експеримент, $Y=f(X1)$, який дозволяє визначити зв'язок між вхідними та вихідними параметрами в умовах дії збурень $X3$ шляхом варіювання факторами за наперед складеним планом.

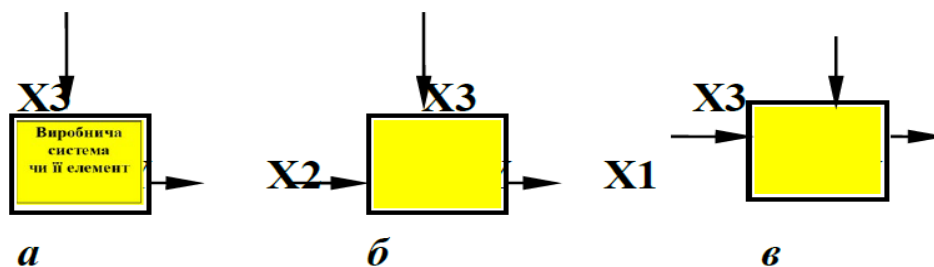


Рисунок 7.2 – Види експериментальних досліджень:
а – статистичний аналіз; б – пасивний; в – активний експеримент

7.2 Статистичний аналіз точності функціонування технологічної системи

Функціонування технологічної машини проходить в умовах дії багатьох випадкових збурень, які викликають певну неточність в реалізації робочих і допоміжних рухів. Як результат виникає розсіяння значень показників якості виробу. При дослідженні процесу функціонування технологічної системи вивчається дія збурюючих факторів на показник її ефективності.

Найбільш відомим прикладом вивчення дії випадкових збурень на процес функціонування технологічної системи є дослідження похибки виготовлення великої партії деталей, коли неможливо передбачити, яке буде відхилення від заданого параметру в конкретній деталі, що обробляється. Часто якість виробу можна охарактеризувати одним визначальним показником, наприклад найточнішим розміром деталі, масою продукту, отриманого при дозуванні, шаром покриття при фарбуванні тощо. Цей показник може використовуватися для характеристики функціональної точності відповідного технічного засобу.

Навіть на добре налагодженому верстаті неможливо передбачити, коли появиться деталь із параметром, що вийшов за межі поля допуску. В цьому випадку можна тільки визначити ймовірність того, що параметр деталі знаходиться в межах поля допуску. Розв'язання проводиться за типовою схемою: знаходиться

теоретичний закон розподілу досліджуваної випадкової величини та вираховується ймовірність прийняття випадковою величиною заданого значення.

Нехай якість виробу, що виготовляється технологічною машиною, або інший вихідний експлуатаційний показник технологічної машини визначаються показником y . Прикладом такого показника може бути найбільш точний розмір при механічній обробці (діаметр валику при шліфуванні, маса продукту, що дозується тощо).

При тривалому функціонуванні машина випустить N виробів. Якщо виміряти в них значення показника y , то виявиться, що його значення мають деяке розсіяння (рис. 7.3).

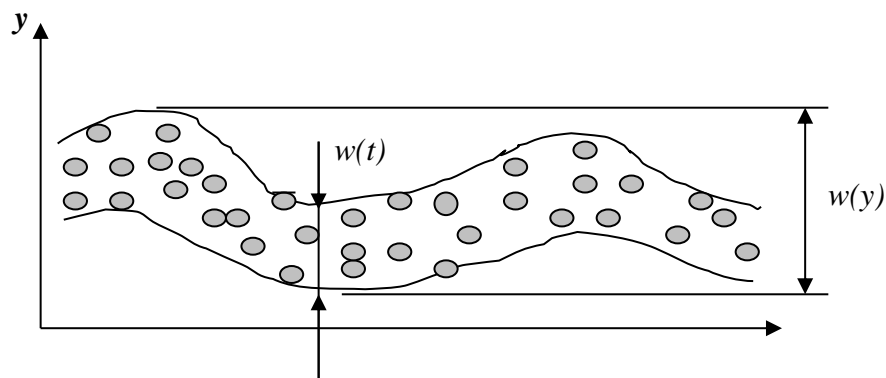


Рисунок 7.3 – Діаграма коливання показника якості виробу при тривалому функціонуванні технологічного комплексу

На діаграмі можна виділити два типи розсіяння – локальне та загальне:

- *локальне розсіяння*, поле якого $w(t)$. Це розсіяння характеризує власну функціональну точність технологічного комплексу або його складових технічних засобів, що визначається конструкцією та внутрішніми збуреннями;

- *загальне розсіяння*, поле якого $w(y)$. Це розсіяння виникає під дією додаткових збурень, які вносяться іншими елементами технологічної системи: коливанням характеристик заготовок, похибками дії оператора при налагодженні та регулюванні, організаційними похибками тощо (рис. 7.4).

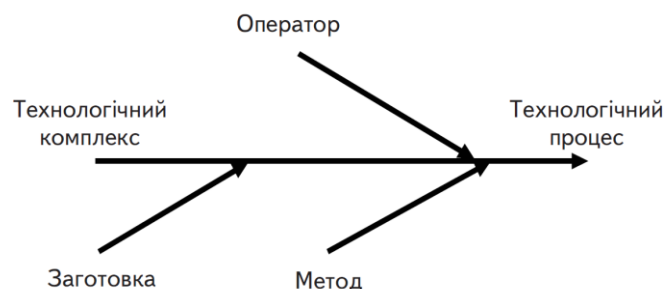


Рисунок 7.4 – Збурення, що впливають на функціональну точність

Функціональна точність по якості виробу оціниться шляхом співставлення параметрів розсіювання із допуском на показник якості реалізації.

Для аналізу функціональної точності по якості виробу технологічного комплексу використовуються наступні показники:

- 1) коефіцієнт точності K_T ;
- 2) коефіцієнт зміщення налагодження K_H .

Коефіцієнт точності визначається як відношення поля розсіювання показника $\omega(y)$ до величини поля допуску на нього T_y за виразом:

$$K_T = \frac{\omega(y)}{T_y}.$$

Коефіцієнт зміщення при налагодженні

$$K_H = \frac{E_H}{T_y} = \frac{|\bar{y} - y_0|}{T_y},$$

де E_H – похибка налагодження, що представляє собою відхилення центру розсіювання y відносно середини поля допуску y_0 (рис. 7.5).

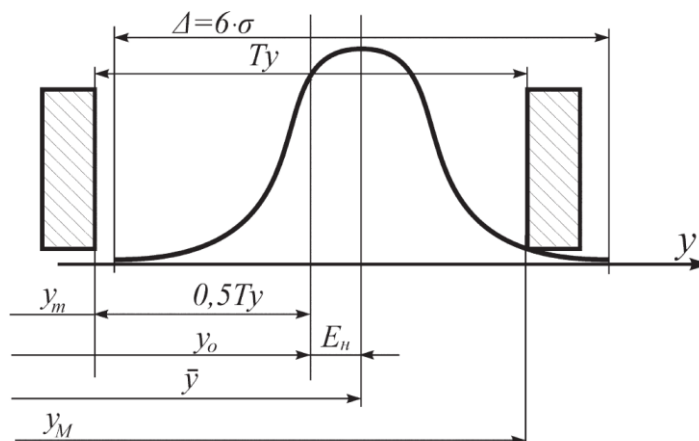


Рисунок 7.5 – Коефіцієнт функціональної точності та коефіцієнт зміщення налагодження технологічного комплексу

Для забезпечення функціональної точності повинні виконуватись умови:

- коефіцієнт точності $K_T < 1$;

• коефіцієнт зміщення налагодження повинен бути менше свого допустимого значення $E_H < [E] = \frac{T_y - w}{2 \cdot T_y}$.

Однак, для визначення цих показників, необхідно знати закон, який описує розподіл значень показника якості. Для цього за результатами експериментальних досліджень визначається відповідність дослідних даних певному типу закону розподілу.

ТЕМА 8 СПРОЩЕННЯ ЕКСПЕРИМЕНТУ МЕТОДАМИ ТЕОРІЇ ПОДІБНОСТІ

Об'єм експериментальної роботи, як було показано раніше, безпосередньо залежить від кількості досліджуваних факторів. Для зменшення кількості незалежних змінних досліджувані фактори та показники досліджуваного процесу чи явища можуть бути згруповані в безрозмірні комбінації, загальна кількість яких буде меншою, ніж їх вихідна кількість. Методика побудови цих безрозмірних комбінацій ґрунтується на аналізі розмірностей змінних, що впливають на досліджуваний процес. Для цього змінні повинні бути незалежні, кількісно визначувані та мати розмірність. Визначимо основні поняття аналізу розмірностей.

Розмірними називаються величини, чисельні значення яких залежать від прийнятих масштабів вимірювання, тобто від системи одиниць вимірювання (довжина, маса, сила, час, енергія тощо).

Безрозмірними називаються величини, чисельні значення яких не залежать від прийнятих масштабів і застосованої системи одиниць вимірювання (кут в радіанах як відношення дуги до радіуса, відношення двох довжин, відношення енергії до моменту сили тощо).

Оскільки всі змінні досліджуваного явища чи процесу як фізичні величини певним чином між собою пов'язані, то частину із цих величин приймають за основні, для них встановлюють одиниці вимірювання, а одиниці решти змінних виражають через одиниці основних. Одиниці вимірювань основних величин називають *основними*, а всі решту – *похідними*. Розмірність похідної одиниці визначається за допомогою визначального рівняння, яке описує її математичний зв'язок з основними одиницями, в якому коефіцієнт пропорційності відповідає одиниці. В загальному випадку вираз для похідної одиниці представляє собою добуток основних величин, піднесених до певного степеню.

До основних одиниць, визначених Міжнародною системою одиниць, які використовуються в механіці, відносяться: кілограм, метр, секунда, градус

Кельвіна. Символ одиниці довжини позначають L , маси M , температури K , часу T . В подальшому розмірність величини x будемо позначати квадратними дужками: $[x]$. Одиниця швидкості, наприклад, визначиться наступним чином

$$[v] = \frac{[l]}{[t]} = \frac{L}{T} = L \cdot T^{-1},$$

де L, T – відповідно розмірності одиниць довжини і часу.

Визначальним рівнянням одиниці сили є другий закон Ньютона

$$F = m \cdot a; [F] = [m] \cdot [a] = M \cdot L \cdot T^{-2}.$$

Розмірність одиниці роботи

$$[A] = [F] \cdot [l] = M \cdot L^2 \cdot T^{-2}.$$

В основу застосування аналізу розмірностей покладено π -теорему подібності, яка формулюється наступним чином:

Якщо існує однорідна відносно розмірностей залежність, що повністю описує процес або явище, наступного вигляду

$$\varphi(y, x_1, x_2, x_3, \dots, x_{n-1}) = 0,$$

яка зв'язує p фізичних величин за допомогою k основних одиниць, то ця залежність завжди може бути поданою як залежність між $(n-k)$ безрозмірними комбінаціями, скомбінованими із цих величин в вигляді

$$\varphi'(\pi_1, \pi_2, \pi_3, \dots, \pi_{n-k}) = 0.$$

В цьому визначенні використано поняття однорідної відносно розмірностей залежності, під якою розуміють рівняння, форма якого не залежить від вибору основних одиниць вимірювання. Наприклад, запис другого закону Ньютона не залежить від вибору одиниць виміру маси рухомого тіла.

Методику розглянемо на прикладі вдосконалення плану експериментального дослідження безконтактного контролю плоских деталей за допомогою пневматичного вимірювального приладу для (рис.7.6, в).

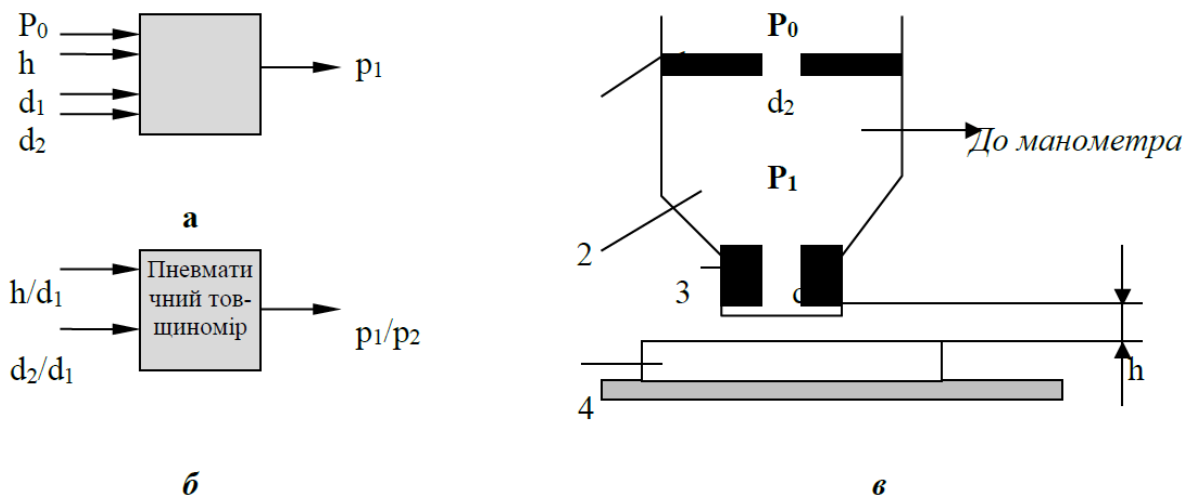


Рисунок 7.6 – Вихідна (а) та вдосконалена (б) схеми експерименту та конструкція товщиноміра (в): 1 – вхідний дросель; 2 – вимірювальна камера; 3 – вихідний дросель; 4 – деталь.

Для проведення вимірювання в вимірювальну камеру приладу через вхідний дросель із прохідним діаметром d_2 поступає стиснене повітря під тиском p_0 . Стиснене повітря виходить через кільцевий зазор висотою h , що утворюється між вихідним дроселем з діаметром d_1 та вимірюваною деталлю. В процесі вимірювань тиск на вході в прилад p_0 підтримується постійним, а тиск в вимірювальній камері p_1 залежить від висоти кільцевого зазору h . Величину тиску p_1 вимірюють, а по його величині роблять висновок про величину відстані від поверхні деталі до вихідного дроселя h , отже, про значення товщини деталі.

Експериментальне дослідження приладу передбачає вивчення впливу на величину тиску в вимірювальній камері p_1 постійного вхідного тиску p_0 , діаметрів вхідного d_2 і вихідного d_1 дроселів, а також відстані h між вихідним дроселем та деталлю (рис.7.6, а). Очевидно, при цьому вважається, що повітря є нестисним, тертя повітряних струменів в приладі відсутнє, інерційні сили при вимірюванні також відсутні. Залежність між цими змінними невідома, тому її запишемо її в загальній формі як

$$p_1 = \varphi(p_0, h, d_1, d_2).$$

В найбільш загальному випадку між цими змінними існуватиме співвідношення вигляду

$$p_1 = \varphi(p_0^a, h^b, d_1^c, d_2^d).$$

Розмірність всіх змінних вихідної залежності може бути виражена за допомогою трьох основних одиниць: маси M , довжини L та часу T (таблиця 1).

Таблиця 1 – Розмірність змінних

Змінна	Позначення	Розмірність
Тиск в вимірювальній камері	p_1	$ML^{-1}T^{-2}$
Вхідний постійний тиск	p_0	$ML^{-1}T^{-2}$
Діаметр вимірювального дросселя	d_1	L
Діаметр вхідного дросселя	d_0	L
Відстань від дросселя до деталі	h	L

Підставимо в вихідну залежність замість змінних їх розмірності

$$ML^{-1}T^{-2} = \varphi[(ML^{-1}T^{-2})^a, L^b, L^c, L^d].$$

Якщо до цього виразу висунути вимогу однорідності відносно розмірностей, як це вимагається в π -теоремі, то показники степені в лівій та правій частинах повинні бути рівними. Тоді отримаємо

$$\begin{aligned} \text{для } M: & \quad 1 = a; \\ \text{для } L: & \quad -1 = -a + b + c + d; \\ \text{для } T: & \quad -2 = -2a. \end{aligned}$$

Ми отримали три рівняння із чотирма невідомими. Виключимо із системи рівнянь величини a і c . Тоді отримаємо

$$1 = a; c = -b - d; 1 = a.$$

Підставивши отримані величини в вихідну залежність, отримаємо

$$p_1 = \varphi(p_0^1, h^b, d_1^{-b-d}, d_2^d).$$

Об'єднаємо члени із однаковими показниками степені, що дозволить перейти до безрозмірних комбінацій змінних:

$$\frac{p_1}{p_0} = \varphi \left(\frac{h}{d_1}, \frac{d_2}{d_1} \right).$$

Такий перехід дозволив спростити експериментальне дослідження, оскільки замість вихідних п'яти змінних в експериментальному дослідженні братимуть участь тільки три безрозмірні комбінації цих величин (рис. рис.7.6, б). При цьому значно спрощується графічне представлення експериментальних даних.

ТЕМА 9 ОФОРМЛЕННЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

Стаття – науковий або публіцистичний твір невеликого розміру в збірнику, або в газеті. Метою статті є викладення нового теоретичного або експериментального матеріалу в даній галузі науки, техніки, економіки тощо.

Наукова публікація – основний, а у фундаментальній науці – практично єдиний результат діяльності вченого. Відповідно, написання публікацій – основне його заняття. Головна мета наукової публікації для автора – зробити роботу надбанням інших дослідників і зафіксувати свій пріоритет у обраній сфері досліджень.

Це досягається публікацією трьох блоків інформації:

- 1) результатів досліджень;
- 2) результатів аналізу;
- 3) повідомлень про себе, як про автора (ів) досліджень і (або) аналізу.

За змістом статті можна розділити на:

- оглядові статті, в яких наводиться інформація про вже опубліковані статті на обговорювану тему, наводиться аналіз стану питання в даній галузі знань, відзначаються тенденції розвитку тощо;
- статті, присвячені вирішенню конкретного питання;
- теоретичні, в яких наводяться нові ідеї, моделі реальних об'єктів, новий погляд на ці об'єкти і т. п.;
- практичні, в яких наводиться опис конкретних методів, пристроїв, технологій, експериментальних даних тощо.

З погляду читача публікація виконує іншу мету. Вона повинна містити короткий, але докладний звіт про проведене дослідження, який дозволяє провести його об'єктивне обговорення. Звіт повинен містити достатню кількість даних і

посилань на опубліковані джерела інформації, щоб колегам можна було оцінити і самим перевірити роботу. Написати хорошу публікацію – означає досягти цих двох цілей.

При всьому різноманітті форм наукових публікацій найбільш важлива з них – стаття в журналі або іншому періодичному чи неперіодичному виданні. Передбачається, що автор, який пише статтю, отримав теоретичні або експериментальні матеріали щодо предмета досліджень.

Предметом (об'єктом) дослідження в техніці можуть бути:

- фізичні ефекти (в тому числі, фізико-хімічні, електрофізичні, електрохімічні і т.п.);
- методи і способи (технології) виробництва продуктів в різних галузях промисловості;
- методи і пристрої контролю параметрів технологічних процесів;
- методи і пристрої для вимірювання різних фізичних величин;
- обладнання, пристрої, прилади, електричні схеми та їх елементи;
- методи створення перерахованих вище технологій і пристроїв;
- обчислювальні алгоритми, інформаційні технології та багато іншого.

Крім того, істотне значення в наукових дослідженнях займають моделі та їх дослідження.

Метою досліджень, як правило, є встановлення суті проблеми (системи, питання тощо), зв'язок цієї проблеми з іншими проблемами, встановлення причинно-наслідкових зв'язків між елементами проблеми, а також між проблемою і зовнішнім середовищем.

Нарешті, будь-яке дослідження – це не лише «задоволення власної цікавості за державний рахунок», а й пошук шляхів вирішення проблеми, шляхів поліпшення стану питання, технічних характеристик тощо.

Результати досліджень, як правило, надаються у вигляді графіків, малюнків, таблиць, моделей, які можуть входити в звіти, дисертації, книги та статті.

Структура статті у науково-технічному журналі регламентується нормативними документами і повинна включати:

- постановку проблеми у загальному вигляді та її зв'язок із важливими науковими чи практичними завданнями (актуальність роботи);
- аналіз останніх джерел і публікацій, в яких описано рішення даної або подібної проблеми і на які посилається автор, виділення невирішених раніше частин загальної проблеми, яким присвячується дана стаття;
- формулювання цілей статті (постановка завдання);
- виклад основного матеріалу дослідження з повним обґрунтуванням отриманих наукових результатів;

- висновки з даного дослідження і перспективи подальших робіт в даному напрямку.

ЗАВДАННЯ ДЛЯ ПЕРЕВІРКИ ЗНАНЬ

1. Поняття про науку як систему знань.
2. Рівні наукового пізнання.
3. Фундаментальні і прикладні дослідження.
4. Особливості розвитку науки.
5. Роль науки в розвитку промислового виробництва.
6. Види прикладних досліджень
7. Методологічні аспекти наукового дослідження.
8. Загальнофілософські методи дослідження.
9. Загальнонаукові методи дослідження.
10. Методи, що використовуються на теоретичному і емпіричному рівнях дослідження
11. Конкретно-наукові методи дослідження.
12. Інформаційне забезпечення наукового дослідження. Інформаційна база досліджень.
13. Способи первинної обробки інформації, її джерела і носії.
14. Наукові документи та видання.
15. Інформаційний пошук.
16. Система УДК, види каталогів.
17. Пошук інформації за ключовим словом.
18. Науково-технічна патентна інформація.
19. Структура МПК.
20. Вибір напрямку та теми, формування задач наукових досліджень.
21. Поняття актуальності дослідження та визначення ступеня його наукової розробки.
22. Попереднє ознайомлення з літературою та визначення головних напрямків дослідження.
23. Збирання і відбір інформації для проведення дослідження.
24. Формулювання загальної і проміжної цілей дослідження.
25. Загальна структура дослідницької роботи .
26. Основні складові і етапи проведення наукового дослідження.
27. Структура фундаментального наукового дослідження.
28. Структура прикладного наукового дослідження.
29. Приклад фундаментального наукового дослідження.

30. Приклад прикладного наукового дослідження.
31. Методологія теоретичних досліджень.
32. Застосування аналітичних моделей і їх агрегування.
33. Застосування системних моделей в теоретичних дослідженнях.
34. Особливості прийняття допущень при побудові теоретичних моделей.
35. Приклад прийняття допущень при побудові теоретичних моделей.
36. Проведення експериментальних досліджень.
37. Розробка програми і плану експериментального дослідження.
38. Визначення послідовності проведення дослідів.
39. Визначення кількості необхідних дослідів.
40. Класичний план проведення однофакторного експерименту.
41. Рандомізований план проведення однофакторного експерименту
42. Метод найменших квадратів.
43. Спрощення експерименту методами теорії подібності.
44. Основи аналізу розмірностей фізичних величин.
45. Основні і похідні величини.
46. Спрощення експерименту методами теорії подібності.
47. Теорема подібності Букінгема.
48. Визначення кількості безрозмірних комбінацій величин.
49. Приклад застосування аналізк розмірностей.
50. Види, особливості викладу та форми впровадження результатів дослідження.
51. Правила оформлення звіту наукової роботи (посилання на джерела, таблиці, ілюстрації, скорочення, формули тощо).
52. Наукова стаття та її структурні елементи.
53. Документи захисту інтелектуальної власності.
54. Відкриття, винахід, вдосконалення в технічній творчості.
55. Правила оформлення патенту на винахід та на корисну модель.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Колесников О. М. Основи наукових досліджень: Київ: Центр навчальної літератури, 2019. 144 с.
2. Мальська М. М, Паньків Н. І. Основи наукових досліджень : навчальний посібник: Львів: Вид-во ЛНУ імені Івана Франка, 2020. 226 с.
3. Основи наукових досліджень: навчальний посібник / О. М. Сінчук, Т. М. Берідзе, М. Л. Барановська, О. В. Данілін, Д. О. Кальмус. Кременчук: ПП Щербатих О. В., 2022. 196 с.

4. Основи наукових досліджень: навч. посіб. / Г. Г. Стрелкова, М. М. Федосенко, А. І. Замулко, О. С. Іщенко : Київ: Вид-во КПІ ім. Ігоря Сікорського, 2019. 120 с.

5. Palchevskiy B., Krestyanpol L. Application of predictive maintenance in the packaging production. *Informatyka, Automatyka, Pomiaru w Gospodarce i Ochronie Środowiska – IAPGOS*. 2022. № 3. P. 27-33.

6. Palchevskiy B., Krestianpol O., Krestianpol L. The use of anylogic system for modelling a flexible automated packing system in training engineering students. *Information Technologies and Learning Tools*. 2020. Vol.75. N 1. P. 225-236.

7. Пальчевський Б. О., Крестьянполь Л. Ю. Побудова діагностичної матриці як компоненти розробки інтелектуальної виробничої системи. *Телекомунікаційні та інформаційні технології*. 2019. №3 (64). С. 63-70.

8. ДСТУ ISO 7870-2:2016(ISO 7870-2:2013, IDT) Статистичний контроль. Карти контрольні. Частина 2. Карти Шухарта - введено в дію 01.09.2016 <https://bitly.su/kfgI>

РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ УПРАВЛІННЯ ТЕХНОЛОГІЧНИМИ КОМПЛЕКСАМИ, ЗАСОБИ ЛЮДИНО- МАШИННОГО ІНТЕРФЕЙСУ

(укладачі Решетило О. М., Смолянкін О. О.)

ТЕМА 1 ОГЛЯД МІКРОПРОЦЕСОРІВ ЇХ КЛАСИФІКАЦІЯ

Програмований контролер – це цифрова обчислювальна операційна електроніка, призначена для використання в промислових умовах. Він використовує програмовану пам'ять для зберігання оперативних вказівок, таких як логічні операції, керування послідовністю, терміни, підрахунки та арифметичні операції, а також контролює різні типи машин за допомогою цифрових та аналогових входів і виходів. процес виробництва. Програмовані логічні контролери та пов'язані з ними периферійні пристрої повинні бути спроектовані таким чином, щоб їх можна було легко інтегрувати з промисловими системами управління та легко розширювати їх функції.

Широкий спектр продуктів ПЛК має різні характеристики та продуктивність. Для PLC класифікація зазвичай базується на різниці в структурі, функціях та кількості входів/виходів.

1.1 Класифікація ПЛК за структурою

Відповідно до структури ПЛК, його можна розділити на два типи: інтегральний тип і модульний тип.

Інтегральний ПЛК.

PLC інтегрального типу інтегрує блок живлення, процесор та інтерфейси вводу-виводу в єдиний шасі, як показано на рисунку 1.1.3 компактною структурою, невеликими розмірами, низькими цінами. Малі PLC зазвичай використовують цю монолітну структуру. Інтегральний ПЛК складається з базової одиниці (також званої хостом) з різними точками вводу-виводу та блоком розширення. Основний блок має процесор, інтерфейс вводу-виводу, порт розширення, підключений до блоку розширення вводу-виводу, та програміст або EPROM. Інтерфейс, до якого підключено маршрутизатор тощо; блок розширення має лише вхід/вихід і блок живлення, а також відсутній процесор. Основні пристрої та блоки розширення, як правило, з'єднуються плоскими кабелями. Інтегральні ПЛК також можуть бути обладнані спеціальними функціональними блоками, такими як аналогові пристрої, блоки управління положеннями тощо, щоб їх функції могли бути розширені.

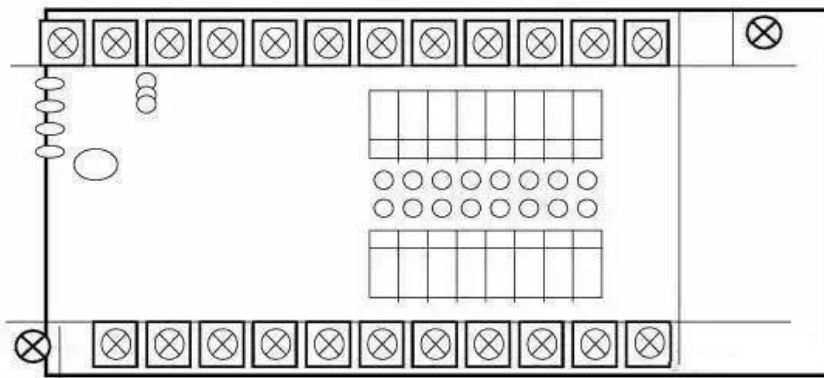


Рисунок 1.1 – Електрична схема інтегрального ПЛК

Модульний ПЛК.

Модульний PLC робить різні компоненти ПЛК окремими модулями, такими як модулі процесора, модулі вводу/виводу, модулі живлення (деякі включені в модуль процесора) та різні функціональні модулі. Модульний ПЛК складається з рами або базової плати та різних модулів, які встановлені на рамі або базовій платі, як показано. Цей модульний PLC характеризується гнучкою конфігурацією та може бути обладнаний системами різних розмірів відповідно до потреб, а також легко збирати та легко розширювати та підтримувати. Великі та середні PLC зазвичай приймають модульну структуру.

Існують також ПЛК, які поєднують в собі інтегральні та модульні функції для формування так званого укладеного (адитивного) ПЛК. Адитивні ЦПУ ПЛК, блок живлення, інтерфейс вводу/виводу тощо також є окремими модулями, але вони з'єднані кабелями, і кожен модуль може бути укладено по шару за шаром. Таким чином, не тільки система може бути налаштована гнучко, але також може бути компактним (рис. 1.2).



Рисунок 1.2 – Типи модульного ПЛК

1.2 Класифікація по функціях

Відповідно до різних функцій ПЛК, його можна розділити на нижні, середні, кінцеві три групи.

Низькочастотний ПЛК

Низькоколіїний ПЛК має такі базові функції, як логічна операція, терміни, підрахунок, переміщення, самодіагностика та моніторинг. Вона також може мати невелику кількість аналогового вводу/виводу, арифметичної операції, передачі даних та порівняння та комунікаційні функції і т. Д. В основному вона використовується для логічного управління та послідовності. Контроль або невелика кількість аналогових керуючих автономних систем керування.

ПЛК середнього класу

На додаток до функцій низькочастотного ПЛК, ПЛК середнього рівня також має багато портів аналогового вводу/виводу, арифметичні операції, передача та порівняння даних, перетворення цифрових систем, віддалені введення / виводу, підпрограми та мережеві зв'язки; деякі також можуть бути додані. Пристрої управління переривань, ПІД-регулювання та інші функції застосовні до складних систем управління.

ПЛК високого класу

На додаток до функцій ПЛК середнього кінця, PLC високого класу також додає арифметичні операції, такі як підписані арифметичні, матричні операції, операції бітової логіки, квадратні кореневі операції та інші спеціальні функції, функції табуляції та функції передачі таблиці. Високоякісний ПЛК має більш універсальну функцію комунікації, яка може бути використана для великомасштабного управління процесом або являє собою систему розподіленої мережі, що забезпечує автоматизацію заводу.

1.3 Класифікація по точках вводу/виводу

Відповідно до кількості точок вводу/виводу ПЛК, його можна розділити на дрібні, середні та великі типи.

PLC маленького розміру.

Платформи невеликого розміру мають менше 256 точок вводу / виводу, однопроцесорні та 8-бітові або 16-бітні процесори, а кількість пам'яті користувача становить 4КВ або менше. Наприклад: серія Mitsubishi FX0S.

ПЛК середнього розміру.

Середні PLC мають точки вводу-виводу від 256 до 2048. Вони мають подвійний процесор і мають користувальницьку пам'ять від 2 до 8 Кбайт.

Великогабаритний ПЛК.

Великі PLCs мають понад 2048 точок вводу-виводу, декілька процесорів, 16-бітових або 32-розрядних процесорів, а кількість пам'яті користувача становить від 8 до 16 Кб.

ТЕМА 2 СТРУКТУРА, ПРИНЦИП РОБОТИ І ПРОГРАМУВАННЯ МІКРОКОНТРОЛЕРІВ ADAM-4000

Контролери та модулі віддаленого введення-виведення серії ADAM-4000.

Контролери серії ADAM-4000 є PC-сумісними контролерами фірми Advantech. Вироби цієї серії є аналогами популярних в Україні контролерів та модулів зв'язку з об'єктом ADAM-4000 (Advantech), NuDAM-6000 (ADLink).

Серія ADAM-4000 забезпечує недороге, гнучке та ефективне рішення для найширшого спектру індустріальних та лабораторних завдань. Вироби цієї серії призначені для управління технологічним процесом, вбудовування в технологічне обладнання, віддаленого збору та обробки інформації, можуть використовуватись як комунікаційні пристрої тощо. Лінійка продуктів включає комунікаційні модулі, модулі аналогового введення і аналогового виведення, дискретного введення/виводу, таймери/лічильники, модулі PC-сумісних контролерів.

Кожен модуль є функціонально-закінченим пристроєм, укладеним у пластмасовий корпус і оснащеним клемними з'єднувачами з гвинтовою фіксацією для підключення входних і вихідних ланцюгів (рис. 2.1). Встановлення модулів не вимагає спеціальних об'єднувальних плат і може здійснюватися як на стандартну DIN-рейку, що несе, так і на будь-яку плоску панель або стінку. Модулі вводу-виводу можуть бути на значній відстані від контролера, підключаючись до нього за інтерфейсом RS-485.



Рисунок 2.1 – Модулі серії ADAM-4000

Загальні технічні характеристики модулів серії ADAM-4000.

- Модулі поєднуються в асинхронну напівдуплексну двопровідну мережу за стандартом RS-485.

- Максимальна довжина сегмента мережі без повторювача – до 1200 м-коду.

- Швидкість передачі даних: 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200 бод.

- Можливість об'єднання до 256 модулів в один сегмент мережі без повторювача.

- Можливість використання в одному сегменті мережі різних швидкостей обміну та форматів даних до $2048 = 256 \times 8$ модулів у системі з використанням повторювачів.

- Формат даних серії ADAM-4000 10 біт = 1 стартовий біт + 8 біт даних + 1 стоповий біт.

- Можливість контролю парності при передачі даних.

- Протокол даних: ASCII символи.

- Напруга ізоляції вхідних ланцюгів 3000 В.

- Живлення від джерела нестабілізованого постійного струму напругою від +10 до +30 В.

- Захист ланцюгів живлення від перенапруги, неправильної полярності підключення живлення.

- Можливість «гарячої» заміни будь-якого модуля.

- Працездатність у широкому діапазоні температур: від -10°C до $+70^{\circ}\text{C}$.

Мережа RS-485 для модулів серії ADAM-4000 є найбільш потужною та гнучкою двопровідною мережею RS-485. Ця мережа працює з різними швидкостями обміну та різними форматами даних. Ця обставина дозволяє об'єднати в одну мережу всі модулі УСО, контролери PLC та інше обладнання, які налаштовані різні швидкості обміну або використовують різні формати даних. У звичайній мережі RS-485 швидкість обміну та формат даних повинні мати однакові значення у всіх пристроїв, підключених до мережі. Перетворювач RS-232 в RS-485 (ADAM -4520) оснащений вбудованим «Self Tuner», що дозволяє йому виявляти швидкість обміну і формат даних автоматично і безпосередньо керувати мережею RS-485.

На рисунку 2.2 представлена розподілена система управління та збору даних на основі модулів ADAM-4000.

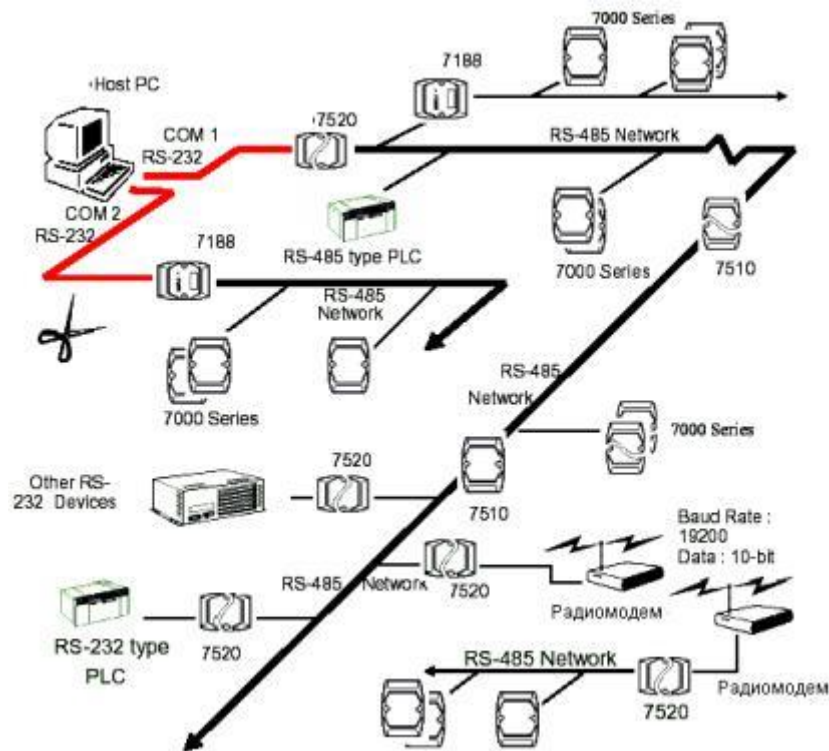


Рисунок 2.2 – Розподілена система управління та збору даних на основі модулів ADAM-4000

Схема взаємодії провідного комп'ютера (HOST) з модулями, об'єднаними в одну мережу на основі RS-485, досить проста. При цьому порядок роботи виглядає так:

1. провідний комп'ютер видає команду в мережу через порт COM1,
2. перетворювач I-7520 перетворює сигнал RS-232 в RS-485,
3. всі модулі, підключені до мережі RS-485, отримують цю команду та порівнюють поле адреси цієї команди зі своєю власною адресою,
4. модуль, у якого ці адреси збігатимуться, виконає цю команду, а інші її проігнорують,
5. після виконання команди модуль посилає відповідь про результати виконання мережу RS-485. Провідний комп'ютер обробляє відповідь та видає наступну команду.

При побудові мережі на основі інтерфейсу RS-485 слід враховувати ту обставину, що лише один із пристроїв у ній може бути ведучим (Master), а інші – веденими (Slave). При цьому під час роботи пріоритети роботи різних пристроїв можуть змінюватися.

Система, побудована на основі модулів серії ADAM-4000, має низку характерних рис:

- простота побудови системи під керуванням HOST-комп'ютера. Модулі поєднуються в мережу будь-якої складної топології на основі широко поширеного

стандарту RS-485; всі модулі використовують для комунікації простий протокол «Запит/Відповідь», який ініціює HOST-комп'ютер,

- «вбудований інтелект». Модулі забезпечують приведення сигналу до необхідного рівня, моніторинг системи, видачу аварійних сигналів, збереження найважливіших параметрів налаштувань,

- гнучкість налаштування. Конфігурація та калібрування модулів здійснюється програмно з HOST-комп'ютера. Параметри конфігурації, такі як адреса, швидкість обміну по послідовному каналу зв'язку, наявність перевірки контрольної суми команди, діапазон зміни вхідних і вихідних сигналів та їх розмірність, вид подання виміряних значень та деякі інші параметри зберігаються у вбудованому ПЗП, що електрично перепрограмується,

- при наявності в мережі модуля I-7188 система може працювати без зовнішнього комп'ютера HOST.

У модулях серії ADAM-4000 реалізовано широкий діапазон швидкостей передачі даних через мережу RS-485 – від 1200 до 115 200 бод. При цьому модулі, підключені до одного послідовного порту HOST комп'ютера, необов'язково повинні працювати на одній і тій же швидкості. Модулі конверторів інтерфейсів RS-232/RS-485, а також модулі повторювачів (ADAM - 4510) мають функцію самоналаштування. Суть її полягає в тому, що ці модулі автоматично визначають напрямок передачі даних, швидкість передачі та формат посилки. Іншими словами, для інформаційних потоків модулі конверторів та повторювачів повністю прозорі.

ТЕМА 3 СТРУКТУРА, ПРИНЦИП РОБОТИ. ПРОГРАМУВАННЯ МІКРОКОНТРОЛЕРІВ ADAM-5000

Тайванська корпорація Advantech утворена у 1983 році і спеціалізується на виробництві обчислювальної техніки для промислової автоматизації. На підприємствах машинобудування, металургії, легкої промисловості багатьох держав добре відомі промислові комп'ютери, програмовані логічні контролери, пристрої збору та передачі даних, що створені спеціалістами цієї фірми. При впровадженні автоматизованих систем управління технологічними процесами широко використовуються засоби автоматизації цієї фірми серій ADAM 4000, ADAM 5000, ADAM 6000, ADAM 8000.

Модулі серії ADAM 4000 є компактними і інтелектуальними пристроями обробки сигналів датчиків і призначені для побудови розподілених систем збору даних і управління на основі інтерфейсу RS-485.

До складу серії ADAM 5000 входять системи розподіленого вводу-виводу даних і управління, а також програмовані контролери. Всі вони

забезпечують до 128 каналів дискретного вводу-виводу або 64 аналогові канали на один блок ADAM 5000. Пристрої серії ADAM 5000 можуть об'єднуватися у багатоточкову мережу на базі інтерфейсу Ethernet або RS-485.

Модулі серії ADAM 6000 призначені для побудови інтелектуальних розподілених систем збору даних і управління на основі інтерфейсу Ethernet. Використання Ethernet дозволяє легко інтегрувати системи на основі модулів ADAM 6000 в мережі Інтернет/інтранет. Здійснюється це шляхом організації Web-доступу до модулів збору даних у реальному часі за допомогою вбудованого в кожен з них Web-серверу. Для організації взаємодії з SCADA-системами верхнього рівня в модулях реалізована підтримка протоколу Modbus/TCP і обмін даними відбувається через OPC-сервер.

Вироби серії ADAM 8000 є новим кроком, зробленим корпорацією Advantech у напрямі освоєння ринку класичних ПЛК-систем. Нове покоління ADAM можуть з успіхом використовуватися у системах промислової автоматизації з підвищеними вимогами до надійності устаткування і до тимчасових параметрів контурів управління. Ці пристрої працюють в промислових мережах MPI, PROFIBUS-DP, ModBus TCP і CAN.

Контролери серії 8000 повністю програмно сумісні з популярними контролерами серії S7-300 фірми Siemens, причому функціональні можливості модулів ADAM 8000 не поступаються оригіналу, а конструктивно нові модулі набагато компактніше.

Серед програмованих логічних контролерів виробництва фірми Advantech найбільш поширеними у промисловості є ПЛК серії ADAM 5000, які складаються з трьох модульних компонентів - процесор, крос плата, модулі вводу-виводу. Кожен пристрій може містити до 4 або до 8 модулів вводу-виводу.

Контролери серії ADAM 5000 умовно можна поділити на дві групи – контролери, що програмуються мовами високого рівня – це контролери групи ADAM 5510, і контролери, програмування яких здійснюється за допомогою

Softlogic-системи KW MULTIPROG мовами міжнародного стандарту IEC 61131-3 – це контролери групи ADAM 5510KW. До першої групи відносяться контролери ADAM 5000, ADAM 5510, ADAM 5510M, ADAM 5510E, ADAM 5510/TCP, ADAM 5510E/TCP, ADAM 5511, а до другої - ADAM 5510KW, ADAM 5510E/KW, ADAM 5510 EKW/TP, ADAM 5510KW/TCP, ADAM 5510EKW/TCP, ADAM 5550. Контролери кожної групи мають багато спільного, різниця між ними визначається тільки об'ємом пам'яті, кількістю слотів та мережним інтерфейсом.

Окрему групу складають контролери ADAM 5510/HC, ADAM 5510/HCG, які мають вбудовану SCADA-систему Trace Mode. Ці контролери зберегли всі

переваги серії ADAM 5000 і придбали додаткові можливості інтегрування систем верхнього і нижнього рівня. Вони підтримують всі функції, що характерні для Trace Mode і їх можна програмувати у тому ж середовищі, що і операторську робочу станцію.

3.1 Мікропроцесорний контролер ADAM 5510

На початку 1998 року фірма Advantech приступила до випуску ADAM 5510, продовжуючи лінію інтелектуальних ПЗО серії ADAM-5000. До серії базових блоків ADAM 5000/485 і ADAM 5000/CAN, орієнтованих на роботу з управляючою машиною верхнього рівня в режимі «питання — відповідь» по одному з популярних промислових інтерфейсів, додався ще один — програмований ADAM 5510.

PC-сумісний програмований контролер ADAM 5510 призначений для використання в локальних розосереджених системах автоматизації у якості автономного контролера. ADAM 5510 забезпечує прийом та видачу аналогових і дискретних сигналів, первинне перетворення їх відповідно до запрограмованих користувачем алгоритмів та обмін інформацією за послідовними каналами зв'язку на базі інтерфейсу RS-485. Мікроконтролер ADAM 5510 має модульний принцип будови, рисунок 3.1. Основою контролера є базовий блок, в який можуть бути встановлені до 4-х модулів вводу-виводу серії ADAM 5000 із загальною кількістю каналів до 64. Габаритні розміри базового блока складають 110×231× 75мм (В×Ш×Г).

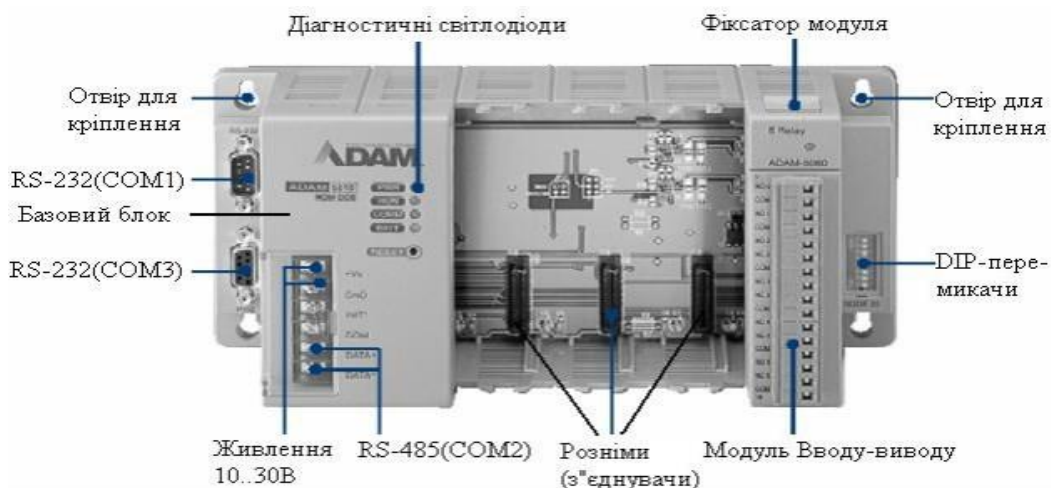


Рисунок 3.1 – Мікропроцесорний контролер ADAM-5510

Базовий блок контролера складається з процесорного модуля, перетворювача напруги, 4-слотової пасивної панелі, трьох послідовних

комунікаційних портів.

Процесорний модуль створений на базі мікропроцесора 80188, який працює під керівництвом операційної системи ROM-DOS на рівні базових функцій, за виключенням BIOS, що дає можливість створення стандартного для IBM-PC програмного забезпечення або прикладних програм, написаних мовою високого рівня "С".

У складі процесорного модуля є флеш-ПЗП об'ємом 512 Кбайт і статичне ОЗП об'ємом 256 Кбайт. Для зберігання прикладної програми, її використання та зберігання даних призначено 170 Кбайт флеш-ПЗП. Крім того, для виконання прикладних програм та пересилання файлів доступні 192 Кбайт оперативної пам'яті.

Функціональна блок-схема контролера приведена на рисунку 3.2.

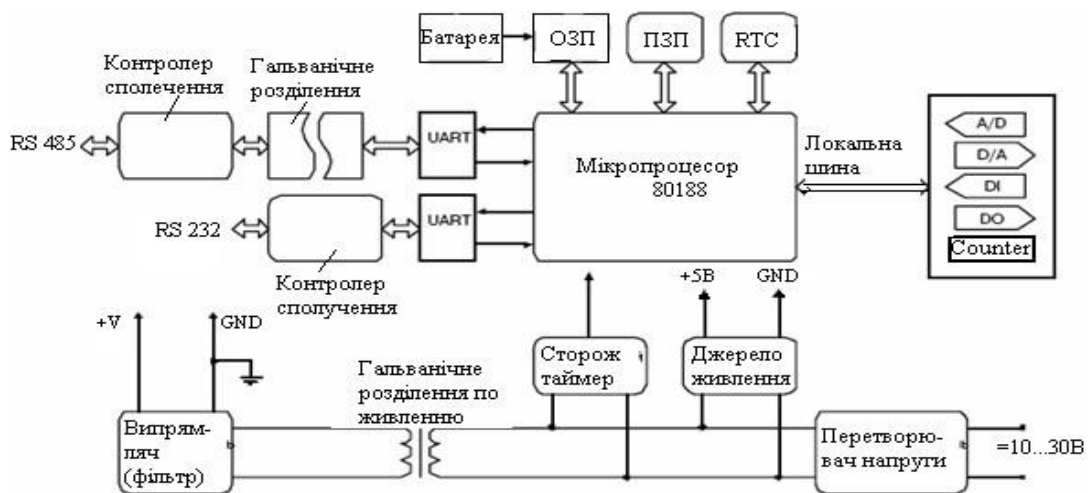


Рисунок 3.2 – Функціональна блок-схема ADAM 5510

Мікроконтролер ADAM 5510 містить три послідовних комунікаційних порти, за допомогою яких можлива взаємодія його з будь-якими пристроями, що мають послідовний доступ. Порт COM1 підтримує роботу в режимі інтерфейсу RS-232, а порт COM2 – роботу в режимі інтерфейсу RS-485. Порт COM3 призначений для здійснення програмування, конфігурування та діагностики контролера ADAM 5510 за допомогою клерувального комп'ютера.

Крім того, він може бути використаним у якості комунікаційного порту загального призначення з інтерфейсом RS-232. Максимальна довжина лінії зв'язку, що обслуговують порти COM1 і COM3 (інтерфейс RS-232) – 15.2м; порт COM2 (інтерфейс RS-485) – 1220м. Швидкість обміну інформацією до 115200 біт/с.

Годинники реального часу, які є у складі мікроконтролера, забезпечують

можливість прив'язки до часу будь-яких подій, що фіксуються у об'єкті керування. Сторожовий таймер призначений для здійснення повторного запуску мікроконтролера у випадку непередбаченого перебою у роботі керувальної програми.

Установлення мережної адреси контролера здійснюється 8-позиційним DIP-перемикачем, який розташовано праворуч на лицевій стороні базового блока. Адреса контролера може мати значення у межах від 0 до 255.

Живлення мікроконтролера ADAM 5510 здійснюється напругою $U=24\text{В}$. Для цього може використовуватися будь-яке джерело з вихідною напругою в діапазоні 10-30В постійного струму.

На лицевій панелі ADAM 5510 розташовані чотири світлодіодних індикатори, які призначені для сигналізації станів блока:

світіння індикатора POWER червоного кольору – є напруга живлення;

світіння індикатора RUN зеленого кольору – йде процес початкового завантаження;

світіння індикатора COMM оранжевого кольору – відбувається обмін за послідовним каналом зв'язку;

світіння індикатора BAT жовтого кольору – розряджений стан батареї.

Модулі вводу-виводу приєднуються до базового блока за допомогою роз'ємів, що закріплені в його слотах. Слоти нумеруються від 0 до 3, а нумерація каналів будь-якого модуля починається з 0. Дискретні модулі забезпечують ввід та вивід дискретних сигналів у діапазоні 10-30В постійного струму, а також релейну комутацію навантажень різної потужності. Модулі аналогового вводу завдяки АЦП перетворюють у цифровий код сигнали з термопар або термометрів опору, а також аналогові сигнали у вигляді напруги та струму. Цифрова форма аналогового сигналу потім перетворюється у формат інженерних одиниць. Модулі аналогового вводу мають оптичну гальванічну розв'язку по вхідних колах і трансформаторну гальванічну розв'язку по колах живлення з напругою ізоляції 3000В постійного струму. Існують аналогові модулі, що мають відповідно три канали вводу сигналів термометрів опору (ADAM-5013), 8 каналів аналогового вводу з диференційними входами (ADAM-5017), 7-каналів аналогового вводу сигналів термопар з компенсацією температури холодних кінців (ADAM-5018), а також 4-канали аналогового виводу (ADAM-5024). Використовуючи сервісне програмне забезпечення, користувач може задати швидкість наростання вихідного сигналу, його початкове значення і тип (струм або напруга), установлення вхідного або вихідного робочого діапазону.

Серед дискретних є 8- (ADAM-5052) і 16- (ADAM-5051) каналні модулі

вводу, 16-канальний модуль виводу (ADAM-5056), вихідні канали якого виконані у вигляді транзисторних ключів, є 6- (ADAM-5060) і 8- (ADAM-5068) канальні модулі з релейними виходами. Існує також 16-канальний модуль дискретного вводу-виводу (ADAM-5050), кожний канал якого може бути незалежно конфігурований в якості вхідного або вихідного за допомогою DIP-перемикачів, що розташовані на платі модуля. Вихідні каскади каналів виконані у вигляді транзисторних ключів за схемою з відкритим колектором, який забезпечує безпосереднє керування малопотужним навантаженням, а при застосуванні реле і керування виконавчими механізмами великої потужності.

Крім того ще існує 4-канальний модуль вводу частотних-імпульсних сигналів (ADAM-5080).

Для підключення зовнішніх пристроїв до модулів контролера використовуються гвинтові клемні колодки. Завдяки тому, що вони роз'ємні суттєво спрощується монтаж проводів і налагоджувальні роботи.

ADAM 5510 має трирівневу гальванічну ізоляцію як для кіл живлення, так і для модулів вводу-виводу з напругою ізоляції 3000В постійного струму, а для портів послідовного зв'язку з напругою ізоляції 2500В постійного струму. Наявність гальванічного розділення дозволяє знизити вплив на систему електромагнітних перешкод, усунути гальванічний зв'язок з електрообладнанням об'єкта керування, а також запобігти несправностей, які можуть з'явитися під впливом випадкових змін напруги в колах живлення, а також перехідними процесами при комутації силового обладнання.

При монтажі мікроконтролер може бути установлений на вертикальну панель або на DIN-рейку.

ТЕМА 4 СТРУКТУРА, ПРИНЦИП РОБОТИ. ПРОГРАМУВАННЯ МІКРОКОНТРОЛЕРІВ МІК-51

4.1 Загальна характеристика та фізична структура

МІК-51 – це компактний малоканальний багатофункціональний мікропроцесорний контролер, призначений для автоматичного регулювання і логічного управління технологічними процесами. Він спроектований, розроблений і вироблений на підприємстві «Мікрол», м. Івано-Франківськ, Україна. Завдяки малоканальності МІК-51 дозволяє економічно управляти невеликим агрегатом або забезпечити високу живучість крупних систем управління.

У контролері передбачено до 9 незалежних контурів регулювання, кожний з яких може бути локальним або каскадним, з аналоговим або імпульсним

виходом, з ручним, програмним (зокрема багатопрограмним) або супервізорним задавачем.

МІК-51 містить засоби оперативного управління, які розташовані на лицевій панелі контролера. За допомогою їх можна вручну змінювати режими роботи, встановлювати завдання, управляти ходом виконання програми, управляти виконавчими пристроями, а також контролювати сигнали і відображати помилки. Стандартні аналогові і дискретні датчики і виконавчі пристрої приєднуються до контролера МІК-51 за допомогою індивідуальних кабельних з'єднувачів. Усередині контролера сигнали обробляються у цифровій формі.

Контролери МІК-51 можуть об'єднуватися в локальну управляючу мережу шинної конфігурації і обмінюватися інформацією. Для такого об'єднання ніяких додаткових пристроїв не потрібно.

Програмування контролера здійснюється за допомогою клавіш передньої панелі або по інтерфейсу за допомогою спеціального програмного забезпечення - візуального редактора АЛЬФА. Система програмування реалізована відповідно до вимог стандарту Міжнародної Електротехнічної Комісії (МЕК) ІЕС 61131-3 і призначена для розробки прикладного програмного забезпечення збору даних і управління технологічними процесами мовою функціональних блокових діаграм Function Block Diagram (FBD).

МІК-51 має розвинені засоби самодіагностики, сигналізації і ідентифікації несправностей. Запрограмована інформація зберігається в ньому при вимкненому живленні завдяки вбудованій енергонезалежній пам'яті.

Контролер МІК-51 є комплексом технічних засобів. У його складі центральний мікропроцесорний блок, клемноблокові з'єднувачі і модулі розширення.

Базова модель контролера має 4 аналогові входи (2 універсальних, 2 уніфікованих), 1 аналоговий вихід, 3 дискретні входи і 5 дискретних виходів.

Модулі розширення мають 7 модифікацій – від МР-51-01 до МР-51-07, які відрізняються кількістю і видом входів-виходів.

Контролер МІК-51 – це проектно-компонований виріб, який дозволяє користувачеві вибирати комплект модулів розширення і клемноблокових з'єднувачів відповідно до кількості і виду вхідних-вихідних сигналів.

Центральний мікропроцесорний блок, рисунок 4.1, має габаритні розміри 96×96×189 (В×Ш×Г) і розрахований на втоплений монтаж у вертикальній панелі щита або пульта управління. Він перетворює аналогову і дискретну інформацію в цифрову форму, обробляє її та формує керувальний вплив.



Рисунок 4.1 – Центральний мікропроцесорний блок контролера МІК-51

МІК-51 має лицеву панель, за допомогою якої можна виконувати оперативне управління контурами регулювання і здійснювати програмування контролера.

Лицева панель в режимі оперативного управління контурами регулювання приведена на рисунку 4.2



Рисунок 4.2 – Лицева панель МІК-51 в режимі оперативного управління

Оперативне управління контурами регулювання здійснюється за допомогою клавіш лицевої панелі, двох чотирирозрядних, одного трирозрядного і одного одnorозрядного цифрових індикаторів, а також набору світлодіодів, які дозволяють змінювати режими, встановлювати завдання, управляти виконавчими механізмами, контролювати сигнали, відображати аварійні ситуації. При програмному регулюванні засоби оперативного управління дозволяють вибирати необхідну програму, пускати, зупиняти, і скидати програму, переходити до наступної ділянки програми, а також контролювати хід виконання програми.

Однорозрядний дисплей «№» показує номер контуру регулятора, з яким

працює оператор. На чотирирозрядний дисплей ПАРАМЕТР виводиться поточне значення регульованого параметра. Два світлодіодні індикатори ▼ і ▲ у цій групі сигналізують про спрацьовування сигналізації виходу параметра, що відображається, за уставки «менше» або «більше». На чотирирозрядний дисплей ЗАВДАННЯ виводиться задане значення регульованого параметру. На трирозрядний дисплей ВИХІД виводиться значення сигналу на виході регулятора. Два світлодіодні індикатори ▼ і ▲ у цій групі сигналізують про спрацьовування виходів імпульсного регулятора в напрямі «менше» або «більше». Чотири світлодіодні індикатори режимів управління вказують, в якому режимі працює контур:

КУ – каскадне управління, ЛУ – локальне управління, РУ – ручне управління, СУ – слідкувальне управління.

Для реалізації оперативного управління контролер МІК-51 має бути переведений у режим РОБОТА і вибрана панель індикації регуляторів (світиться індикатор РЕГ).

Оперативне управління ведеться за виборчим принципом: вибирається номер контуру і для нього контролюються і змінюються режими, параметри і сигнали. Виключення складає лише контроль помилок, який ведеться для всього контролера в цілому.

Всі зовнішні кола підключаються до контролера МІК-51 через 3 розніми, що розташовані на задній стінці приладу. Через верхній рознім підключаються кола аналогових і дискретних входів - виходів, кола живлення дискретних виходів і інтерфейс.

Через середній рознім підключаються модулі розширення МР-51.

Через нижній двоконтактний рознім-клему підключається електроживлення контролера МІК-51.

Кабелі, що з'єднують контролери МІК-51 з датчиками і виконавчими механізмами, приєднуються до них через клемно-блокові з'єднувачі КБЗ-29-01(КБЗ-29Р-01, КБЗ-29К-01, КБЗ-29С-01) (рис. 4.3), або КБЗ-24-10, які виконані на основі безгвинтових плоско-пружинних клем і встановлюються на рейку DIN35x7.5 EN50022.



Рисунок 4.3 – Клемноблокові з'єднувачі

4.2 Архітектура контролера

Архітектура контролера МІК-51 описує інформаційну організацію контролера і характеризує його як ланку системи управління. Частина архітектурних елементів структури реалізована апаратно, частина - програмно. Все програмне забезпечення, що формує архітектуру, зашите в постійному запам'ятовуючому пристрої і користувачеві недоступно. Незалежно від того, які реалізовані елементи архітектури – апаратні або програмні – користувач може уявляти контролер як виріб, в якому всі елементи реально існують у вигляді окремих вузлів.

До складу архітектури контролера входять:

- апаратура вводу-виводу інформації (базової моделі і модулів розширення);
- апаратура оперативного управління і настройки;
- апаратура інтерфейсного каналу;
- бібліотека функціональних блоків;
- функціональні блоки.

Апаратура вводу-виводу інформації розрахована на прийом і видачу двох видів сигналів – аналогових і дискретних. Формування імпульсних сигналів на виході імпульсного регулятора виконується програмно і ці сигнали поступають на виконавчі механізми через дискретні виходи контролера. Апаратура вводу (вхідні ПЗО) перетворює аналогові і дискретні сигнали, що поступають на вхід контролера, в цифрову форму, а апаратура виводу (вихідні ПЗО) здійснює зворотне перетворення. Всі аналогові і дискретні входи і виходи контролера повністю універсальні. У початковому стані вони не «прив'язані» до яких-небудь обмежень контролера. Така прив'язка здійснюється користувачем і реалізується в процесі програмування.

Апаратура оперативного управління (лицева панель) розрахована на оператора-технолога і оператора-наладника. За допомогою набору клавіш, цифрових і світлодіодних індикаторів оператор-технолог «веде» технологічний процес - контролює його параметри, змінює режими управління та уставки, пускає, зупиняє, скидає програму і т.ін.

За допомогою лицевої панелі можна програмувати контролер, виконувати настроювання його параметрів, а також контролювати сигнали у внутрішніх точках структури. У загальному вигляді лицева панель можна розглядати як інтерфейс з людиною.

Апаратура інтерфейсного каналу забезпечує послідовний зв'язок декількох контролерів. Цей канал має приємопередавач, який перетворює вхідний потік послідовних біт інформації у паралельний код представлений у вигляді байтів,

а також здійснює зворотне перетворення. Завдяки достатньо великій швидкості передачі побітної інформації через інтерфейсний канал, вважається, що всі сигнали передаються одночасно.

Бібліотека функціональних алгоритмів достатня для того, щоб можна було розв'язувати порівняно складні задачі автоматичного регулювання і логіко-програмного управління. У ПЗП «зашиито» більше 50 типів функціональних алгоритмів безперервної і дискретної обробки інформації, включаючи блоки ПД регулювання, функціональні блоки математичних, динамічних, нелінійних, аналого-дискретних і логічних перетворень.

Частина бібліотечних алгоритмів виконує особливу задачу – вона зв'язує апаратуру контролера з основною масою функціональних блоків. До цих «зв'язних» функціональних алгоритмів відносяться:

- функціональні алгоритми вводу-виводу аналогових і дискретних сигналів;
- функціональний алгоритм обслуговування панелі користувача;
- функціональні алгоритми прийому і передачі сигналів через інтерфейсний канал.

Апаратні елементи архітектури контролера (вхідні і вихідні ПЗО, лицева панель, інтерфейсний канал) починають виконувати свої функції лише після того, як будуть задіяні відповідні функціональні блоки.

Існує біля 99 блоків з вільним заповненням їх будь-якими функціональними алгоритмами бібліотеки і вільною конфігурацією між собою та входами-виходами контролера. У загальному випадку функціональний блок має свої входи, виходи, параметри настройки і функціональне ядро. Входи функціонального блока з'єднуються сполучними лініями з іншими блоками. Один або декілька функціональних блоків, що сполучені зв'язками між собою, утворюють FBD-програму. Функціональним блокам у програмі користувача присвоюються порядкові номери, які ідентифікують блок в системі, а також визначають черговість виконання блоків у програмі.

У загальному випадку бібліотечний функціональний блок має наступні реквізити:

- бібліотечний номер, який відповідає номеру функції, що виконується;
- номер алгоблока;
- модифікатор розміру;
- базова адреса реєстрової області параметрів;

Бібліотечний номер є двозначним десятковим числом і основним параметром, що характеризує властивості функціонального блока.

Модифікатор звичайно задає кількість однотипних входів або операцій, які

може виконувати функціональний блок. Наприклад, в суматорі модифікатор розміру задає кількість сумованих входів, а в програмному задавачі – кількість ділянок програми і т.ін. Модифікатор вказується для функціональних блоків, в яких він є, і не вказується для блоків, в яких модифікатор відсутній.

Базова адреса функціонального блоку – це номер комірки регістрової області пам'яті контролера в якій повинен розміщуватися перший параметр цього функціонального блоку. Параметри кожного функціонального блоку розміщуються в регістровій області один за одним, починаючи з адреси 0000, вказаної у відповідному реквізиті функціонального блоку.

Базова адреса регістрової області параметрів функціонального блоку настроюється користувачем в режимі програмування на рівні конфігурування функціонального блоку [7].

Кожний функціональний блок графічно має вигляд прямокутника, усередині якого є позначення функції, яка виконується блоком, модифікатор розміру, номер алгблока і його параметри.

Програмна модель функціонального блоку приведена на рисунку 4.4.

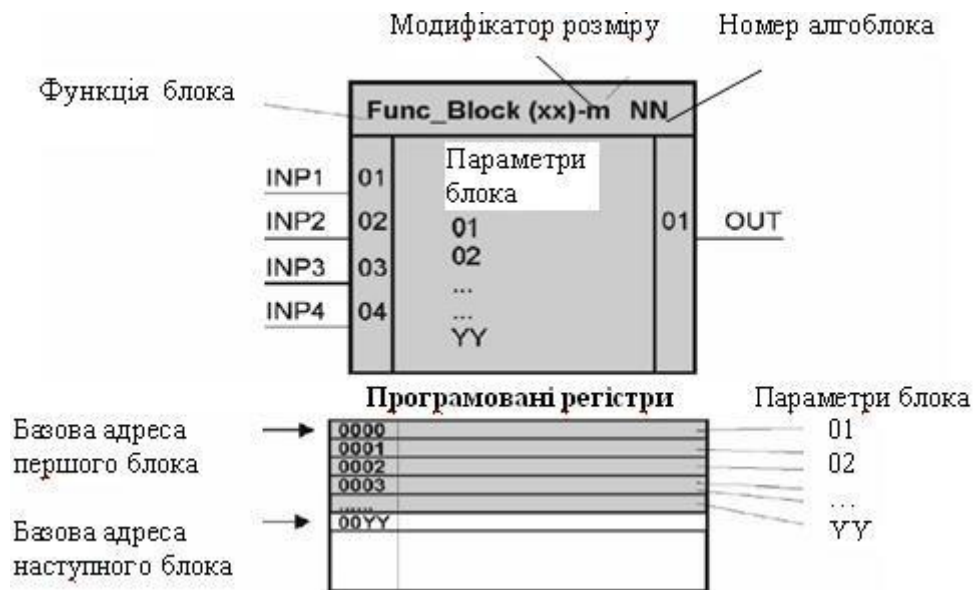


Рисунок 4.4 – Програмна модель функціонального блоку

Кожен блок залежно від виконуваної їм функції має певну кількість входів і виходів. Входи (INP) завжди розташовані зліва, а виходи (OUT)-справа.

Параметри настройки функціонального блоку задають йому різні властивості, наприклад, номер вхідного сигналу, коефіцієнт посилення, масштабний коефіцієнт, зсув сигналу, постійна часу фільтру, максимальні і

мінімальні межі зміни будь-якого параметра і т.ін. Параметри настройки функціонального блока відрізняються від входу блока тим, що параметр настройки блока може залишатися не підключеним.

Перелік сигналів, які обробляються функціональними блоками, включає наступні типи:

- аналоговий (вимірювані параметри, значення аналогових входів-виходів, технічні одиниці, коефіцієнт, швидкість зміни параметра і т.ін.);
- дискретний, імпульсний;
- числовий;
- часовий (уставка часу, тривалість імпульсу).

Аналогові сигнали формуються на виході таких функціональних блоків, як регулятори, суматори, задавачі, інтегратори і т.ін. До аналогових сигналів відносяться такі параметри настройки, як межі спрацьовування компаратора, рівень обмеження і т.п. Не дивлячись на те, що на аналогових входах і виходах контролера сигнал змінюється в діапазоні 0-100 % на виході функціональних блоків аналоговий сигнал може змінюватися в ширшому діапазоні – в діапазоні відповідних типів даних. Це дозволяє, наприклад, скласти два числа, кожне з яких дорівнює 90 %, і на виході суматора одержувати правильний результат. Якщо результат яких-небудь обчислень виявляється більше встановленого типу даних, то сигнал на виході функціонального блока обмежується значеннями відповідного типу даних.

Часові сигнали формуються на виході таймерів, програмних задавачів, одновібраторів і подібних функціональних блоків. До часових сигналів відносяться такі параметри настройки, як постійні часу, протяжність часу, час витримки і т.п.

Числові сигнали – це сигнали на виході лічильника або інших функціональних блоків, робота яких пов'язана з відліком подій. Числовими можуть бути і параметри настройки, наприклад, число може задати граничне значення сигналу на виході лічильника, номер етапу, до якого повинна перейти логічна програма і т.п.

Дискретні сигнали звичайно обробляються логічними функціональними блоками і функціональними блоками, пов'язаними з перемиканням сигналів. Проте дискретними можуть бути і параметри настройки. Наприклад, дискретні сигнали у функціональному блоці завдання визначають, чи повинне виконуватися статичне і динамічне балансування.

Масштабний коефіцієнт – це параметр настройки функціональних блоків, в яких потрібне масштабування сигналів. Так, цей коефіцієнт використовується у функціональних блоках аналогового вводу і

виводу, підсумовування з масштабуванням і т.ін. Коефіцієнт пропорційності використовується в основному у функціональних блоках регулювання.

Швидкість зміни аналогових сигналів – це параметр настройки, що задає, наприклад, швидкість зміни при динамічному балансуванні або задає обмеження швидкості у відповідному функціональному блоці.

Тривалість імпульсу – це параметр настройки функціонального блока імпульсного виводу. Цей параметр задає мінімальну тривалість імпульсу, який сформовано імпульсним регулятором. Тривалість імпульсу відлічується в десятих частках секунди.

Технічні одиниці – це параметри настройки функціональних блоків оперативного контролю. За допомогою цих параметрів задається формат числа, в якому контрольовані параметри (наприклад, параметр, завдання, неузгодження і т. ін.) виводяться на індикатори лицевої панелі.

В результаті виконання блоком відповідної функції на його виходах формуються сигнали, які визначаються станом вхідних сигналів і значенням параметрів, що настраюються. Наприклад, якщо на вхід функціонального блока суматора поступає чотири сигнали, то ці сигнали підсумовуються з урахуванням відповідних масштабних коефіцієнтів, які вказані в параметрах і результуючий сигнал поступає на вихід функціонального блоку.

Кількість входів і виходів функціонального блока не фіксовані і визначаються видом функціонального блока. Ні в одному функціональному блоці число входів і виходів не перевищує 12. Деякі функціональні блоки мають неявні входи або виходи, які не доступні для конфігурування. До таких функціональних блоків відносяться функціональні блоки вводу-виводу, функціональні блоки приємопередачі, функціональний блок панелі користувача.

Як тільки один з таких функціональних блоків використовується, його неявні входи або виходи автоматично з'єднуються з апаратурою, яку вони призначені обслуговувати.

Наприклад, коли використовується функціональний блок аналогового вводу, його неявні входи автоматично з'єднуються з АЦП відповідного каналу. А на виходах цього функціонального блока будуть сформовані загальнодоступні сигнали, еквівалентні сигналам, що поступають на аналогові входи контролера. Тому, якщо на вхід будь-якого функціонального блока потрібно подати аналоговий сигнал, цей вхід при програмуванні необхідно з'єднати з відповідним виходом (масштабованим або не масштабованим) функціонального блока аналогового вводу AIN (рис. 4.5).

Неявні входи і виходи на графічних схемах функціональних блоків не відображаються. Всі входи і параметри функціонального блоку мають скрізну

нумерацію від 1 до 99. Виходи функціонального блока також нумеруються двузначною десятковою цифрою від 1 до 99.

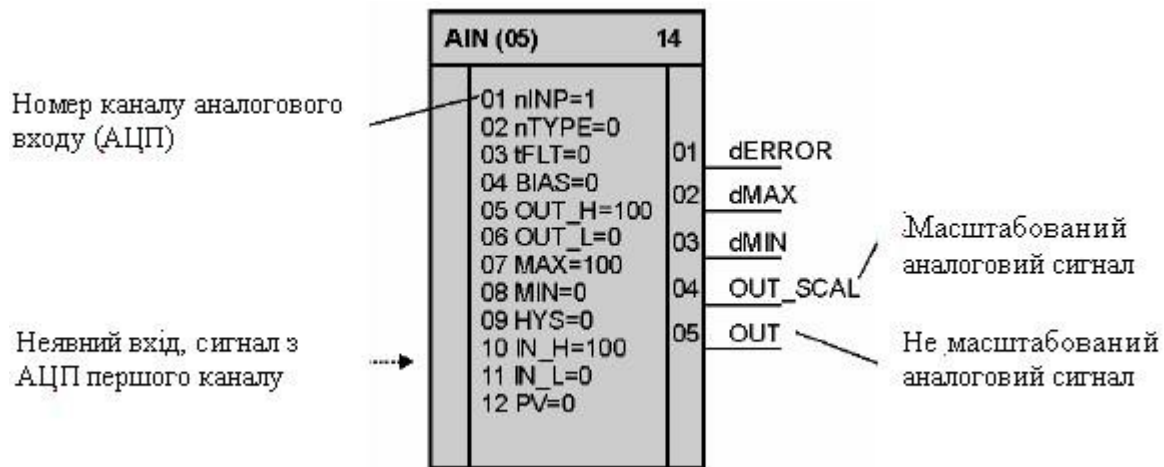


Рисунок 4.5 – Функціональний блок аналогового вводу з неявними входами

Якщо параметри функціонального блока є зв'язаними параметрами, тобто підключеними до яких-небудь виходів інших блоків, то їх нумерація здійснюється із зсувом, рівним максимальному номеру входу.

При програмуванні мовою функціональних блоків в більшості випадків діють правила:

- функціональним блокам привласнюються порядкові номери, які ідентифікують блок в системі, а також визначають черговість виконання блоків в програмі;
- функціональному блоку може бути привласнений будь-який порядковий номер;
- в одній програмі функціональні блоки одного і того ж типу можна використовувати багато разів;
- не може бути вільних (непідключених) входів функціонального блоку;
- виходи функціонального блока можуть залишатися вільними (непідключеними);
- допускаються сполучення входів і виходів будь-якого типу, оскільки перетворення типів здійснюється автоматично.

З цих правил є наступні виключення:

- для деяких функціональних блоків є обмеження на кратність їх використання в межах одного контролера. Так, функціональний блок аналогового вводу можна використовувати лише чотири рази – ці чотири функціональні блоки конфігурують всі аналогові входи. Аналогічні обмеження (з аналогічних причин) мають інші функціональні блоки вводу-

виводу інформації.

При конфігуруванні для входів задається номер функціонального блоку і номер виходу, до якого підключається даний вхід.

Всі можливості конфігурування однакові як для входів, так і для параметрів.

Зв'язок параметрів функціональних блоків дозволяє виконати автоматичну зміну будь-якого параметра настройки (автопідстроювання).

Сигнал на будь-якому вході при необхідності можна інвертувати. Для безперервних сигналів інверсія означає зміну знаку, а для дискретних - зміна стану (заміну 1 на 0 і 0 на 1).

Інвертування дозволяє, наприклад, віднімати сигнали за допомогою суматора, запускати або скидати таймер не переднім, а заднім фронтом сигналу і т.ін.

Можливості конфігурування не залежать від функціонального блока і визначаються наступними правилами.

Обслуговування функціональних блоків здійснюється послідовно за циклічною схемою. Спочатку обслуговується перший функціональний блок, потім другий і т.д. поки не буде обслужений останній функціональний блок. Коли час в межах встановленого циклу закінчиться, програма знову перейде до обслуговування першого функціонального блока. Час циклу виконання програми користувача фіксований і складає 0,1с. Якщо контролер включений в мережу, тоді залишок часу циклу після обробки програми використовується для обробки запитів, отриманих по інтерфейсному каналу. Мережний обмін носить випадковий характер по відношенню до циклу контролера. Якщо контролер не встиг завершити обмін інформацією в одному циклі, він допрацьовує розпочатий обмін у наступних циклах.

Загальний час, що витрачається на обслуговування функціональних блоків ТФБ і інтерфейсного каналу ТІН, повинен бути менше ніж час циклу, тобто:

$$ТФБ + ТІН < 0,1 \text{ с.}$$

У кожному циклі функціональний блок одержує на свої входи сигнали, що обчислені в попередньому циклі функціональними блоками, до яких даний функціональний блок підключений по конфігурації.

Циклічність обслуговування функціональних блоків призводить до того, що затримка в обробці сигналів залежить від порядку програмування сполучених між собою функціональних блоків. Цю обставину слід враховувати при програмуванні функціональних блоків і для мінімізації затримки бажано, щоб функціональний блок – джерело мав менший номер, ніж функціональний

блок-приймач (величина цієї різниці значення не має). Коли, наприклад, номер блока-джерела менше за номер блока-приймача, затримка складає один цикл, а коли номер блока-джерела більше за номер блока-приймача, затримка складає два цикли.

В контролері МК-51 все різноманіття типів сигналів і параметрів настройки обслуговується загальноприйнятим набором типів даних. При цьому забезпечується простий інтерфейс з базами даних інших мов програмування, що сприяє уніфікації і розвитку програмного забезпечення контролера.

4.3 Мережна архітектура

Контролер МК-51 має інтерфейсний канал послідовного зв'язку RS-485, за допомогою якого контролери можуть об'єднуватися в локальну або розподілену управляючу мережу. У мережі контролери можуть обмінюватися інформацією, як з комп'ютером, так і між собою. Всього в мережі може бути до 31 контролера, у тому числі інших моделей.

Контролери в мережі зв'язуються один з одним за допомогою витої пари проводів. Довжина лінії зв'язку може досягати 1200 м (на одному сегменті мережі) залежно від встановленої швидкості передачі даних. Для збільшення довжини лінії зв'язку, а також кількості сегментів – використовують магістральні підсилувачі.

Мережа має багатоточкову конфігурацію і шинну топологію. Для побудови мережі контролерів, призначених для обміну інформацією з комп'ютером, необхідний додатковий пристрій, який використовується для перетворення інтерфейсів RS-232 в RS-485 - БП-485 (або USB в RS-485 - БП-52).

ЗАВДАННЯ ДЛЯ ПЕРЕВІРКИ ЗНАНЬ

1. Що таке програмований логічний контролер (PLC) і для чого він використовується в промислових умовах?
2. Які основні компоненти складають програмований логічний контролер?
3. Як працює пам'ять в PLC і які типи даних вона може зберігати?
4. Які типи входів і виходів підтримує PLC і як вони взаємодіють з зовнішніми пристроями?
5. Які основні логічні операції виконує PLC?
6. Як PLC може бути використаний для керування послідовністю операцій у виробничому процесі?

7. Що таке аналогові та цифрові входи/виходи в контексті PLC і чим вони відрізняються?
8. Як PLC може бути інтегрований з іншими промисловими системами управління?
9. Які переваги програмованих логічних контролерів перед традиційними релейними системами управління?
10. Як розширити функціональність PLC для складніших завдань керування?
11. Які існують методи програмування PLC і як вони впливають на його роботу?
12. Які засоби безпеки передбачені в PLC для запобігання аварійним ситуаціям?
13. Які новітні тенденції в розвитку програмованих логічних контролерів?
14. Які основні характеристики серії ADAM-4000 роблять її ефективним рішенням для індустріальних та лабораторних завдань?
15. Які типи модулів входять до серії ADAM-4000 і які завдання вони виконують?
16. Які переваги використання RS-485 інтерфейсу для підключення модулів введення-виведення серії ADAM-4000?
17. Як модулі серії ADAM-4000 інтегруються з іншими промисловими системами управління?
18. Як забезпечується безпека та надійність передачі даних у системах з використанням модулів ADAM-4000?
19. Які прикладні області можуть використовувати модулі серії ADAM-4000 для віддаленого збору та обробки інформації?
20. Чому можливість розташування модулів на значній відстані від контролера є важливою в промислових застосуваннях?
21. Як модулі аналогового введення і виведення серії ADAM-4000 використовуються для управління технологічними процесами?
22. Які особливості комунікаційних модулів серії ADAM-4000 дозволяють їм виконувати роль ефективних комунікаційних пристроїв?
23. Які типи входів і виходів підтримують модулі дискретного введення/виведення серії ADAM-4000?
24. Як модулі серії ADAM-4000 можуть бути використані для створення систем моніторингу та контролю в реальному часі?
25. Які основні відмінності між модулями серії ADAM 4000, ADAM 5000, ADAM 6000 та ADAM 8000?
26. Які переваги надає використання програмованих контролерів серії ADAM 5000 в системах розподіленого вводу-виводу?

27. Як реалізована підтримка до 128 каналів дискретного вводу-виводу або 64 аналогових каналів в одному блоці ADAM 5000?

28. Як мережеві можливості модулів серії ADAM 6000 сприяють їх інтеграції в інтернет/інтранет через Ethernet?

29. Що таке вбудований Web-сервер у модулях ADAM 6000, і як він використовується для організації Web-доступу до даних у реальному часі?

30. Як модулі серії ADAM 6000 підтримують протокол Modbus/TCP для взаємодії зі SCADA-системами верхнього рівня?

31. Які особливості роблять модулі серії ADAM 8000 придатними для використання в промислових автоматизаційних системах з підвищеними вимогами до надійності?

32. Чим відрізняється конструкція модулів ADAM 8000 від їх аналогів, і які переваги це забезпечує?

33. Як серія ADAM 8000 працює з промисловими мережами, такими як MPI, PROFIBUS-DP, ModBus TCP, і CAN?

34. Які основні функції виконує мікропроцесорний контролер МІК-51 у технологічних процесах?

35. Які переваги надає малоканалність контролера МІК-51 при управлінні невеликими агрегатами?

36. Як у МІК-51 реалізовані 9 незалежних контурів регулювання, і які можливості вони надають?

37. Що таке каскадне регулювання, і як його можна застосувати в контролері МІК-51?

38. Які типи виходів (аналогові або імпульсні) підтримує контролер МІК-51, і як вони впливають на його роботу?

39. Які режими роботи передбачені в МІК-51, і як ними можна керувати за допомогою оперативних засобів на лицевій панелі?

40. Як реалізоване ручне та програмне управління в контролері МІК-51?

41. Як аналогові і дискретні датчики підключаються до МІК-51, і які типи з'єднувачів використовуються?

42. Які основні функції та призначення системи UltraLogik32 у розробці програмного забезпечення для збору даних та управління?

43. Як стандарт ІЕС 61131 впливає на структуру і функціональні можливості UltraLogik32?

44. Які мови програмування підтримує система UltraLogik32 і в яких випадках використовується кожна з них?

45. Що таке Function Block Diagram (FBD) і як ця мова використовується в UltraLogik32 для розробки програмного забезпечення?

46. Яким чином UltraLogik32 підтримує візуальне програмування і чому це важливо для розробників?
47. Які переваги надає апаратно-незалежна платформа UltraLogik32 з інваріантним компілятором?
48. Як система UltraLogik32 інтегрує зовнішні програмні модулі, написані на інших мовах, таких як Сі, Паскаль, Асемблер?
49. Що таке мова ST (Структурований текст) і які переваги вона надає для програмування в UltraLogik32?
50. Які можливості надає мова релейних діаграм (Ladder Diagrams, LD) у системі UltraLogik32?
51. Як система UltraLogik32 забезпечує апаратну незалежність при розробці програмного забезпечення для IBM PC-сумісних контролерів?
52. Як система UltraLogik32 забезпечує взаємодію з промисловими комп'ютерами із відкритою архітектурою?
53. Як виглядає зовнішня програма, створена за допомогою UltraLogik32, у вигляді функціонального блоку?
54. Як UltraLogik32 реалізує підтримку побудови і детального опису алгоритмів управління технологічними процесами?

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Пупена О. М., Ельперін І. В. Програмування промислових контролерів у середовищі UnityPro: навч. посібник. Київ: Ліра-К, 2019. 376 с.
2. Ніколаєнко А. М., Пупена О. М., Ельперін І. В. Програмування промислових контролерів. Конспект лекцій. Запоріжжя, 2020. 100 с.
3. Галкін П. В., Ключник І. І. Програмування ПЛК в CODESYS: навчальний посібник. Харків: ФОП Панов А. М., 2019. 92 с.
4. Пупена О. М., Ельперін І. В., Луцька Н. М., Ладанюк А. П. Промислові мережі та інтеграційні технології в автоматизованих системах. Навчальний посібник. Київ: Ліра-К, 2019. 500 с.
5. Програмування промислових контролерів. Конспект лекцій для студентів напряму 151 «Автоматизація та комп'ютерно-інтегровані технології» денної та заочної форм навчання / А. М. Ніколаєнко, О. М. Пупена, І. В. Ельперін. Запоріжжя: Запорізький національний університет, 2020. 100 с.

ІНТЕЛЕКТУАЛЬНІ МЕТОДИ УПРАВЛІННЯ ТА ЦИФРОВІ ТЕХНОЛОГІЇ

(укладач Сацик В. О.)

ТЕМА 1 ПОНЯТТЯ ІНТЕЛЕКТУАЛЬНИХ МЕТОДІВ, СИСТЕМ ТА ТЕХНОЛОГІЙ

1.1 Загальні відомості

Сучасний етап розвитку цивілізації міцно пов'язаний із стрімким технологічним зростанням та розбудовою інформаційного суспільства, у якому завдяки широкому використанню інформаційно-комунікаційних технологій суттєво збільшується інтенсивність інформаційного обміну, а основним типом діяльності стає обробка інформації та генерування нового знання.

Вказані чинники стають визначальними у розвитку економіки, науки, освіти. Обумовлена цими обставинами комп'ютеризація діяльності підприємств та організацій, взаємозалежність технологій веде до постійного вдосконалення автоматизованих інформаційних систем і технологічних засобів. Протягом останніх років спостерігається стійка тенденція інтелектуалізації систем і технологій, збільшення кількості мобільних пристроїв, що розширюють доступ до інформації, а також поява нових методів та алгоритмів щодо її опрацювання.

Завдяки сучасним темпам розвитку інформаційних технологій усе більш широке використання у різних сферах діяльності знаходять та постійно вдосконалюються системи підтримки прийняття рішень. Враховуючи існуючі можливості прикладних додатків та засобів моделювання, інтерес до систем підтримки прийняття рішень (СППР) як до перспективного напрямку забезпечення підтримки прийняття рішень і потужного інструментарію підвищення ефективності праці у сфері управління безперервно зростає.

Увага до систем класу СППР особливо зростає, адже в них зосереджена особливо цінна аналітична та консолідована інформація. Крім того, зацікавленість користувачів загострюється й наявністю в цих системах передових математичних моделей і методів, унікальних алгоритмів та програмних засобів. У таких умовах ці системи повинні вміти задовільняти широкому спектру вимогам користувачів на різних щаблях управління, забезпечувати взаємодію як з внутрішніми фахівцями, так із зовнішніми, підтримувати як локальні рішення, так і глобально скоординовані. Одним з важливих напрямів вдосконалення таких систем є їх інтелектуалізація.

Проблемам створення засобів інтелектуалізації систем, інструментарію аналітичної обробки інформації, що циркулює в них, приділяється серйозна увага

науковців і фахівців, завдяки чому у цій царині напрацьовано чимало теоретичних робіт та отримані значні практичні результати.

Виходячи з викладеного, а також враховуючи, що підготовка фахівця за спеціальностями «Автоматизація та компютерно-інтегровані технології» є дуже важливою, роль та значення дисциплін, пов'язаних з питаннями інтелектуальної обробки та аналізу інформації в автоматизованих системах серед інших дисциплін, що вивчаються, набуває також виняткового значення.

Усе живе на Землі обмінюється інформацією і несе у своїх генах знання. Людина як розумна істота навчилася сприймати та зберігати інформацію і знання в різних формах. Спочатку була усна традиція, потім з'явилася писемність. Об'єм інформації і знань почав нестримно збільшуватися. У ХХ ст. люди навчилися передавати знання майбутнім поколінням не лише на папері, але й в цифровому (електронному) виді. Потoki інформації, як видно з рисунка 1.1, почали зростати в геометричній прогресії.

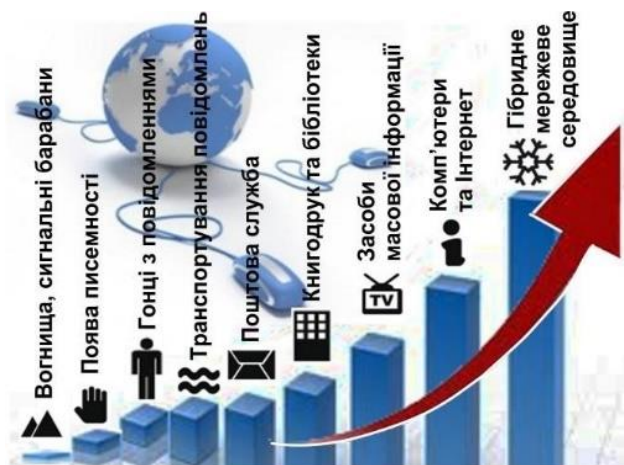


Рисунок 1.1 – Зростання об'ємів інформації у суспільстві

Водночас, з розширенням індустріалізації, збільшенням виробництва головним елементом економічної структури країн стали підприємства (організації, установи). Саме тут сформувалися процеси управління (менеджмент) та прийняття рішень, які стали важливими і визначальними в досягненні ефективності їх діяльності.

Із-за значних об'ємів даних та взаємозв'язків між ними знайти відповіді на питання в цьому інформаційному просторі стало дуже важко, а в деяких предметних областях фактично й неможливе. Для розуміння людиною такої предметної області виникла необхідність отримати додатковий інструмент, роль якого зараз з успіхом відіграють комп'ютери та автоматизовані інформаційні системи.

Перші покоління автоматизованих систем були спрямовані в основному на обробку даних та розв'язання обчислювальних та облікових задач (бухгалтерія, складський облік). З часом автоматизовані системи почали використовувати вже для управління підприємствами, технологічними процесами виробництва тої чи іншої продукції. Основою цих систем стали бази даних (БД), які забезпечували структурування інформації, її зберігання у великих об'ємах та інтеграцію окремих функціональних задач управління у єдиний комплекс.

Поступово потреби досягнення усе більш ефективних цілей управління почали вимагати створення нових інструментальних засобів, які дозволяють скоротити витрати, що неминуче виникають завдяки обмеженням людських можливостей в обробці числових даних, аналітичному опрацюванні інформації та швидкості виконання осмислених операцій. Тому ще з початку 70-х років ХХ ст. розпочалися розробки засобів автоматизованої підтримки прийняття управлінських рішень, у результаті чого склалися умови створення й успішного використання нових людино-машинних систем – систем підтримки прийняття рішень.

З початком ХХІ століття набула поширення концепція постіндустріального суспільства, що отримало назву інформаційного суспільства, як нової історичної фази розвитку цивілізації, в якій головними продуктами виробництва є інформація і знання. Цей щабель в піднесенні сучасної цивілізації характеризується збільшенням ролі інформації і знань в житті суспільства, зростанням долі інфокомунікацій, інформаційних продуктів і послуг у валовому внутрішньому продукті, створенням глобального інформаційного простору.

Оці тенденції, що диктували необхідність знаходження механізмів «трансформації» інформації в знання й використання цього знання як ресурсу для підтримки прийняття управлінських рішень, в сфері автоматизованих інформаційних систем знайшли відображення в процесах інтелектуалізації систем й інформаційних технологій.

Нові виклики інформаційного суспільства пов'язані з електронним веденням бізнесу, інтелектуальними методами управління технологічними процесами впровадженням технологій електронного урядування, коли обмін документами та повідомленнями відбувається в електронній формі. Якщо слідувати ідеям відомого канадського публіциста Дона Тапскотта, глобальність змін, що відбуваються з людством, визначає напрям «цифрова економіка» чи «епоха мережевого інтелекту», якими вона рухатиметься в перспективі, або, як різні фахівці іменують, «ера інформації», «цифрова епоха», «економіка, заснована на знаннях». Логіка Тапскотта будується на авторській схемі, що включає п'ять елементів, які складають ланки цифрового суспільства, де кожен наступний елемент включає

попередні, утворюючи при цьому нову якість. Вищезгадана схема (ланки цифрового суспільства) представлена на рисунку 1.2.



Рисунок 1.2 – Ланки цифрового суспільства (електронно-цифрове співтовариство)

Таким чином, майбутня цифрова спільнота буде пов'язаною з інтелектуалізацією економіки та технологій, де поряд з проблемою управління інформацією буде вирішуватись і проблема «керування знаннями».

1.2 Інтелектуалізація систем

Що ж таке інтелектуалізація інформаційних систем? Щоб відповісти на це запитання треба з'ясувати, у чому полягають особливості традиційних автоматизованих інформаційних систем (АІС), для цього необхідно розглянути поняття інформаційного процесу, який представлено на рисунку 1.3. Будь-яка комп'ютерна інформаційна система зазвичай реалізує той самий інформаційний процес, виконуючи наступні функції: сприймає інформаційні запити розв'язання задачі (1) і необхідні початкові дані, обробляє запити і дані (2), що зберігаються в системі, у відповідності з відомим алгоритмом і формує необхідну вихідну інформацію (3).

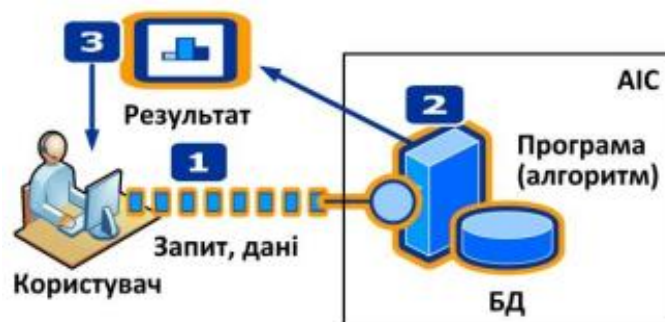


Рисунок 1.3 – Інформаційний процес, що реалізується комп'ютерною системою

У цьому процесі обробляється знання про предметну область. Це знання має двояку природу. Відомості про об'єкти предметної області, що накопичуються в звичайній базі даних, представляють так зване фактуальне знання. Залежності і відношення між об'єктами предметної області, які дозволяють інтерпретувати відомості (факти) з БД або витягати з них інформацію, відображають операційне знання. Операційне знання представляється в алгоритмічній формі як правила перетворення даних (у вигляді програми, що виконується комп'ютером).

Така схема організації інформаційного процесу має суттєві недоліки. У разі виникнення змін в предметній області (а це відбувається постійно) фактуальні знання ще можна внести в систему шляхом актуалізації БД. А операційне знання скорегувати дуже важко (фактично треба переписувати програму, що оперативно зробити неможливо). Водночас для формулювання запиту користувач має ясно уявляти собі структуру БД і до певної міри алгоритм розв'язання задачі. Отже, користувач повинен досить добре бути обізнаним не лише в проблемній області, а й в логічній структурі БД і алгоритмі програми, що висуває додаткові вимоги до користувача. У випадку багатокористуцткої системи ця проблема взагалі є нерозв'язаною.

Таким чином, вказані загальні недоліки традиційних інформаційних систем полягають в слабкій адаптивності до змін в предметній області та інформаційних потреб користувачів, в неможливості розв'язувати задачі, що погано формалізуються, з якими управлінські працівники постійно мають справу.

Перераховані недоліки усуваються в інтелектуальних інформаційних системах завдяки виділенню операційного знання з програми у так звану базу знань, яка в декларативній формі зберігає загальні для різних задач одиниці знань (правила). При цьому структура, що цим управляє, набуває характеру універсального механізму розв'язання задач (машини висновку), який зв'язує одиниці знань у виконуваних ланцюжки (генеровані алгоритми) залежно від конкретної постановки задачі (сформульованої в запиті мети і початкових умов). В наслідок цього такі інтелектуальні системи отримали назву систем, основаних на знаннях. Нарешті, такі засоби дозволяють організувати діалог з користувачем на мові, наближеної до природньої та спримати заздалегідь неформалізовані запити (природно-повний інтерфейс, (ППІ)). Таким чином інформаційний процес в інтелектуальних системах трансформується до виду, приведену на рисунку 1.4.

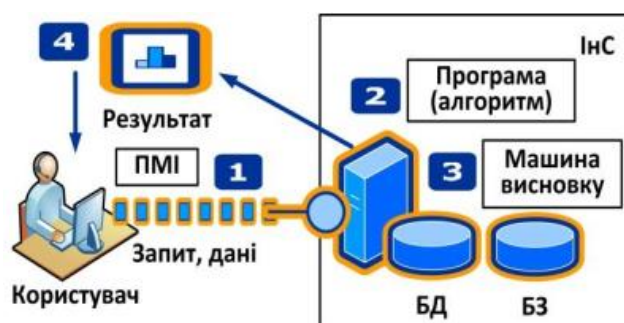


Рисунок 1.4 – Інформаційний процес, що реалізовується інтелектуальною системою

Отже, можна навести такі ознаки інтелектуальних систем:

- підвищення «грамотності», або «освіченості» системи за рахунок накопичення знань;
- можливість подавати та використовувати знання про знання (метазнання);
- вміння розв'язувати складні задачі, що важко формалізуються;
- можливість опрацювання неточних, неясних, невизначених знань;
- адаптивність, тобто здібності розвитку та пристосування;
- можливість отримання нових знань з тих, що є в наявності (властивість самонавчання);
- комунікативність, пов'язана з можливістю формулювання користувачем довільних запитів до системи на мові, максимально наближеної до природньої.

В наш час процеси інтелектуалізації інформаційних систем знаходять вираз у створенні найрізноманітніших систем і технологій, орієнтовну класифікацію яка наведена на рисунку 1.5. З цими системами познайомимося докладніше на подальших сторінках даного розділу, але загальним в них є те, що вони є комплексом програмних, лінгвістичних і логіко-математичних засобів для реалізації основного завдання: забезпечення підтримки розумової діяльності людини, зокрема в сфері підтримки прийняття рішень, інформування і консультування. Важливим напрямком є й сфери імітації інших властивостей людини, що знаходить вираз у створенні роботів. На рисунку 1.5 відповідно: САПР – системи автоматизації проектування, САНД – системи автоматизації наукових досліджень.



Рисунок 1.5 – Класифікація інтелектуалізованих систем

У чому ж переваги інтелектуалізації систем у порівнянні з традиційними інформаційними системами, зокрема для різних підприємств та установ? Відповідь на це питання надає рисунок 1.6.



Рисунок 1.6 – Переваги інтелектуалізації систем

Зрозуміло, що усі переваги привести в одному рисунку майже неможливо, адже загальна ефективність інформаційної системи визначається ступенем відповідності її складових виконанню системою свого призначення (функцій) згідно із критерієм цілі. А в кожній системі, як правило, свій перелік функцій і власне

визначення цілі, які залежать від галузі застосування та вимог конкретних користувачів.

1.3 Штучний інтелект як основа інтелектуалізації

У чому ж основа засобів інтелектуалізації комп'ютерних систем? Її пов'язують з відносно новим науковим напрямком – теорією штучного інтелекту. Термін *artificial intelligence* (AI), що в нашому перекладі й означає «штучний інтелект» (ШІ), вперше був запропонований 1956 року Джоном Так-Карті (John McCarthy) на семінарі з аналогічною назвою, присвяченому розробці методів розв'язання логічних задач, який відбувся в Дартмутському коледжі в США. Не можна сказати, що цей термін є вдалим, адже дві складові – «штучний» і «інтелект», взагалі кажучи, принаймні на сьогоднішній день, протилежні один одному. Тим не менш він виявився життєздатним і увійшов у широкий обіг як практиків так і науковців. Але ж з тих часів вчені і фахівці поділені на два табори – на тих, що вірять в можливість створення штучного інтелекту, й тих, хто піддає теорію ШІ нищівній критиці.

Дослідження в області моделювання процесу мислення розпочалися задовго до появи комп'ютера. З кінця 40-х років минулого століття виділилися два напрями моделювання, які тривалий час були практично незалежними, а саме логічне і нейро-кібернетичне.

Перше ґрунтується на виявленні і застосуванні різних логічних і емпіричних прийомів (евристик), які застосовує людина для вирішення яких-небудь завдань. Другий напрям – нейро-кібернетичний – спирається на побудову систем, що самоорганізуються, та складаються з множини елементів, функціонально подібних до нейронів головного мозку.

Прихильники цих течій відверто проголошують, що машина може мислити. Видатний англійський математик і філософ Алан Тюрінг, щоб замінити безглузде, на його думку, запитання «чи може машина мислити?» на більш визначніше, запропонував 1950 року в статті «Обчислювальні машини і розум» (*Computing machinery and intelligence*) спеціальний тест. Замість того, щоб абстрактно сперечатися про критерії, які дозволяють відрізнити живу мислячу істоту від машини, що виглядає як мисляча, він запропонував спосіб встановити те, що може бути реалізованим на практиці, як це представлено на рисунку 1.7. Суддя-людина С обмежений час (наприклад, декілька хвилин) переписується на природній мові з двома співрозмовниками, яких він не бачить (А і В). Один з них – людина, а інший – комп'ютер. Якщо суддя С за наданий час за отриманими відповідями від А і В не зможе надійно визначити, хто є хто, то вважається, що комп'ютер пройшов тест

Тюринга, тобто його можна визначити як інтелектуальну істоту. До цього часу, строго кажучи, жодна комп'ютерна система тест Тюринга не пройшла.

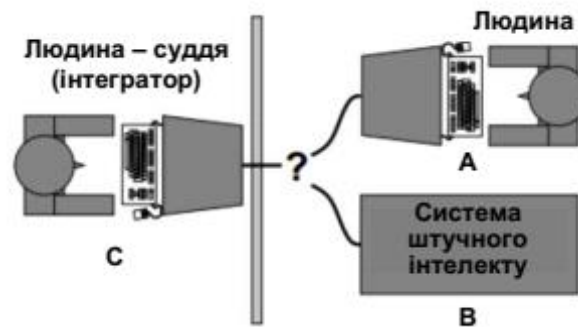


Рисунок 1.7 – Тест Тюринга на інтелектуальність

Але провідний фахівець Google Р. Курцвейл передбачає, що до 2029 року комп'ютерний мозок зможе пройти цей тест, який доведе наявність у нього розуму в людському розумінні цього слова.

До речі, тест Тюринга несподівано знайшов практичне застосування. Компанія Yahoo для захисту від атак спам-ботів (програм-роботів, що розповсюджують рекламні повідомлення шляхом реєстрації на сайтах в Інтернеті) реалізувала ідею застосувати те, що легко дається людям, але представляє труднощі для комп'ютерів: розпізнати в процесі реєстрації спотворені літери, що важко читаються. Люди в змозі розшифрувати їх і ввести правильний текст за лічені секунди, але програма буде збитою з пантелику. Це творіння було назване Captcha (*Completely Automated Public Turing Test to Tell Computers and Humans Apart* – повністю автоматизований публічний тест Тюринга для розрізнення комп'ютерів і людей), і воно зараз повсюдно використовується в Інтернеті.

Багато ранніх досліджень в сфері штучного інтелекту здійснювалися на основі таких поширених настільних ігор, як шашки, шахи, «п'ятнадцятки», хрестики – нулики. На додаток до притаманного їм «інтелектуального» характеру, такі ігри мають деякі властивості, що роблять їх ідеальним об'єктом для експериментів. Більшість ігор ведуться з використанням чітко визначеного набору правил: це дозволяє легко будувати простір пошуку і позбавляє дослідника від неясності і плутанини, властивих менш структурованим проблемам.

Тривалий час дослідження щодо пошуку можливостей створення штучного розуму проводилися у напрямку імітації на комп'ютері гравця у шахи як ігри, що зазвичай вважається найінтелектуальнішою. На цьому шляху були досягнуті значні успіхи. Достатньо згадати серію матчів 1996-1997 рр. колишнього чемпіона світу з шахів Г. Каспарова проти комп'ютера Deep Blue, який створила фірма ІВМ. Ця система була здатна аналізувати 200 млн. ходів у секунду, або 50 млрд. позицій

протягом трьох хвилин, виділених на один хід у шаховій партії. Спочатку Каспаров вигравав, застосовуючи нетрадиційні (парадоксальні) ходи. IBM сильно оновила Deep Blue, поліпшила її логіку, і пізніше комп'ютер виграв у матчі з б ігор (дві перемоги, три нічі).

З цього приводу видатний Г. Саймон зазначив: «Усі наші експерименти з шаховими програмами показали, що висока швидкість обчислень не замінює знання, хоча між ними й є певний взаємозв'язок. Що цікаво і чудово в людях простих або складних, як вам буде завгодно – це наскільки ми вибіркові через те, що не можемо робити розрахунки з такою величезною швидкістю, ми не робимо багато обчислень за хвилину (більшість з нас і за декілька хвилин). Але ми замінюємо цю здатність своєю вибірковістю, ґрунтованою на знаннях, – ми йдемо одним правильним шляхом, а не пробуємо усі варіанти, які можливі. Це й є наш розум. Ось у чому його секрет».

Отже, усе, що вміли комп'ютери – швидко перебирати варіанти ходів («я сюди – він сюди; – я сюди – він сюди»), відкидаючи свідомо невдалі варіанти. Але 2016 року відбулася подія, що суттєво змінила ставлення багатьох до моделювання ігор на комп'ютері. Цього року був організований матч Лі Седоля, одного з найсильніших професійних гравців світу в гру го, і комп'ютера AlphaGo, створеного фахівцями з різних країн. Через чотири години і тринадцять хвилин після початку матчу схвильований Лі Седоль збирав камені з дошки, оголосивши третю підряд поразку від комп'ютера, і, тим самим, поразку в п'ятиматчевій серії.

Справа у тому, що гравці в «Го» завжди вважали свою гру неймовірно складною для «кремнієвого розуму». Кількість гілок ходів в го більше кількості атомів у Всесвіті – особливо не перебереш. Більше того, в шахах вже відіграні усі дебюти, що допомагає скоротити довжину перебору, а в го кількість варіацій на початку партій також є приголомшливо великою, причому постійно з'являються нові ідеї і ходи. Оцінюють, що за час існування го (а це тисячі років), не було зігране двох однакових партій!

Як вдалося комп'ютеру виграти у такій складній грі? Творці AlphaGo об'єднали найуспішніші підходи, накопичені в індустрії штучного інтелекту: нейронні мережі глибокого навчання, генетичні алгоритми (копії AlphaGo, що трохи відрізняються, грали самі з собою і відбиралися кращі) і методи Монте-Карло. В комп'ютер завантажили півтори сотні тисяч партій (десь 30 мільйонів людських ходів), у верхніх шарах нейромережі утворилися деякі абстрактні міркування «збалансованості ігрової дошки», а потім AlphaGo змусили грати саму з собою, щоб розвинути ці міркування, тобто самонавчатися.

У результаті експертам, що спостерігали за грою здавалося, що боротися з Лі Седолем вийшов не програмний код, а ним бореться істота, що має накопичену

пам'ять, здатність до прогнозування ігрової ситуації і досвід перемоги в десяти мільйонах матчів проти таких же істот, що не пройшли еволюційного відбору. AlphaGo грало як дуже сильна людина. Іншими словами, цей комп'ютер пройшло тест Тюринга! Якщо AlphaGo запустити на ігровий го-сервер KGS в Інтернеті, ніхто не зможе розкрити, що грає не з людиною. «Істота», що явно має якісь ідеї, обходить людину в грі, що є невід'ємною частиною людської культури, при цьому людиною явно не є!

Незважаючи на такі приголомшливі результати, все ж таки поки що штучний інтелект є специфічним і не автономним, адже, хоча AlphaGo розуміє гру «Го» краще за людину, він нічого більше не вміє. Копії AlphaGo для гри з самим собою створюються і відбираються алгоритмом, написаним людиною. Також AlphaGo не може (поки що?) самостійно створити середовище для відтворення самого себе. Крім того, ідеї AlphaGo неможливо вилучити з нього і перекласти людською мовою, тобто шари нейрмережі, що утворилися у наслідок «міркування» комп'ютера, представляють «чорний ящик», і це може мати певні ризики у випадку інших застосувань подібних реалізацій.

Отже, поки вчені б'ються над створенням мислячого комп'ютера, в рамках наукового напрямку штучного інтелекту народжуються практичні засоби інтелектуалізації інформаційних систем. Нижче наведені деякі з основних напрямків ШІ та його застосувань, що впливають на розвиток інтелектуалізації систем, а саме:

- 1) представлення знань та маніпуляція знаннями,
- 2) комунікації «людина-машина»,
- 3) розпізнавання образів,
- 4) навчання та самонавчання,
- 5) планування дій, пошук розв'язків задач,
- 6) самоорганізація, методи евристичної самоорганізації,
- 7) автоматизація створення програм.

Зокрема такі засоби створюються на основі теорій нейромереж, генетичних алгоритмів, еволюційного моделювання, нечіткої логіки та ін.

1.4 Інтелектуальні технології

Отже, перетворити інформаційні системи на інтелектуальні дозволяють інтелектуальні інформаційні технології, які на сьогодні є однією з найбільш перспективних і таких, що швидко розвиваються наукових і прикладних сфер інформатики. До основних інтелектуальних технологій необхідно віднести: обробка текстів на природній мові, моделювання знань і бази знань, управління знаннями, розпізнавання образів, нейротехнології, математичні моделі,

концептуальне програмування та ін., а також відповідні інструментальні засоби створення інтелектуальних систем (ІНС). Ознайомимося з деякими з цих технологій.

Центральними для ІНС є технології моделювання та представлення знань про предметні області. На основі сучасного уявлення про категорії знань розроблені базові моделі подання знань в комп'ютерних системах, такі як фрейми, семантичні мережі, правила та ін., що дозволило розробити концептуальні основи технології баз знань.

Роботи в області нейрокібернетики створили основу технологій нейронних семіотичних систем. Розроблені на основі нейротехнологій різні нейропакети використовуються для моделювання сенсорних і мовних систем людини штучними нейронними мережами.

Технології автоматичного розпізнавання образів є перспективними для ІНС. На сьогодні її реалізації використовуються переважно для оптичного читання текстів.

Взагалі автоматизації роботи зі знаннями, представленими в текстовому виді приділяється багато уваги. Гіпертекстові моделі і системи, методи витягання знань з тексту і комп'ютерного пошуку в тексті, технології автоматичного реферування і анотування текстів, машинного перекладу і автоматичної класифікації документів знаходять усе більше застосування. Особливий інтерес викликають комплексні інтелектуальні програмні засоби для обробки текстів.

Важливе значення мають гіпертекстові системи, призначені для реалізації пошуку за ключовими словами у базах текстової інформації. Інтелектуальні гіпертекстові системи відрізняються можливістю використання складнішої семантичної організації ключових слів, яка відбиває різні смислові відношення термінів. Таким чином, механізм пошуку працює передусім з базою знань ключових слів, а вже потім безпосередньо з текстом.

Інтелектуальні бази даних відрізняються від звичайних баз даних можливістю вибірки за запитом користувача необхідної інформації, яка може явно не зберігатися, а виводитися з даних, наявних у базі. При цьому ця технологія намагається здійснити пошук за умовою, яка може бути довідзначеною в ході розв'язання задачі. Така система без допомоги користувача по структурі бази даних сама буде шлях доступу до необхідних даних. Формулювання запиту здійснюється в діалозі з користувачем, послідовність кроків якого виконується в максимально зручній для користувача формі.

В інтелектуальних системах особливе значення має реалізація інтелектуального інтерфейсу, коли запит користувача формулюється за допомогою висловлювань на природній мові. Для реалізації природно-мовного інтерфейсу

необхідно вирішувати завдання морфологічного, синтаксичного і семантичного аналізу на основі внутрішньо-машинного рівня представлення знань.

Інтерфейс користувача з ІнС за допомогою графічних образів, які генеруються відповідно до подій, що відбуваються, дозволяють здійснювати технології когнітивної графіки. Графічні образи в наочному і інтегрованому виді описують множину параметрів ситуації, що розглядається.

Системи управління знаннями базуються на технологіях сховищ даних і інтелектуального аналізу даних (*Data mining based on Data Warehouse*). На відмінність від інтелектуальної бази даних інформаційне сховище є сховищем з можливістю витягувати значиму інформацію з БД на основі оперативного аналізу даних. Для вирішення цих завдань потрібно застосування методів витягання знань з баз даних (*Knowledge Discovery*), що ґрунтуються на застосуванні методів багатовимірного статистичного аналізу, індуктивних методів побудови дерев рішень, нейронних мереж, генетичних алгоритмів тощо.

Сучасний стан робіт в області метаданих для інформаційних ресурсів спрямований зокрема на інтелектуалізацію Інтернету (концепція семантичного веба). Напрацьовані стандарти і специфікації, становлять ядро платформи ХТЛ, що служить технологічною основою семантичного веба.

Технологія створення програмного забезпечення для інтелектуальних систем істотно відрізняється від розробки традиційних програм з використанням поширених мов програмування. Трудомісткість розробки ІнС значною мірою залежить від інструментальних засобів, що використовуються. Ці засоби включають спеціальні мови програмування, орієнтовані на обробку символічної інформації (мова LISP, орієнтована на обробку списків, SMALLTALK, мова рекурсивних функцій РЕФАЛ), мови логічного програмування (PROLOG), мови представлення знань (OPS5, KRL, FRL), інтегровані програмні середовища, що містять арсенал інструментальних засобів для створення систем, а також так звані оболонки експертних систем (BUILD, EMYCIN, EXSYS, Professional), які дозволяють створювати прикладні експертні системи без програмування. Останнім часом термін «оболонка» вживається все рідше, його замінюють ширшим поняттям «середовище розробки».

Недоліком спеціальних мов програмування є слабка пристосованість до об'єднання з програмами, написаними на мовах традиційного програмування. Одним з сучасних засобів, що дозволяє використовувати низку підходів, які забезпечують підтримку програмування на основі правил об'єктно-орієнтованого і процедурного програмування є мова CLIPS. Назва цієї мови – аббревіатура від C Language Integrated Production System. Тожливості логічного висновку і

представлення знань у вигляді правил і фактів зробили CLIPS ефективним засобом розробки експертних систем.

1.5 Коротка історична довідка

Історію інтелектуалізації систем доцільно розпочати з появи самого поняття «система» (організм, лад, союз, ціле, складене із частин), що виникло в Древній Греції близько 2000 років тому, та робіт Аристотеля, Демокрита, Декарта, Платона й ін., що утворили основу системного аналізу. Суттєвий поштовх у становленні цього поняття зробили розвиток астрономії (Коперник, Галілей, Ньютон й ін.), а також філософські погляди, теорії про структуру пізнання (Бекон, Гегель, Ламберт, Кант, Фіхте й ін.).

Діяльність людини з формування техносфери, пов'язані з цим розвиток виробництва, машин і техніки, промислова революція ХІХ ст. та індустріальна епоха стали причиною появи поглядів натуралістів ХІХ-ХХ ст. (Богданов, Бергаланфі, Вінер, Ешбі, Цвіккі й ін.).

На початку ХХ ст. уперше виникла думка про те, що не атоми, а інформація є справжньою основою всього суцього. Багато в чому саме завдяки таким поглядам ми тепер можемо повномасштабно фіксувати і розраховувати матеріальні і нематеріальні аспекти існування і діяти відповідно до них. Поглянувши на світ з точки зору інформації-безкрайніх просторів даних, людство отримало небувале представлення про навколишню дійсність.

Як наслідок заговорили про «науку управління», про прийняття рішень, виникли системи планування, технічної підготовки виробництва. Корінний перелом у цьому паприку був пов'язаний із появою комп'ютера (1946) та кібернетики (робота Вінера 1948 р.).

Свій внесок зробила й подальша структуризація виробництва, виокремлення технологічних операцій, координація виробництва і пов'язані із цим теоретичні дослідження з планування, статистики, прийняття рішень. В цілому передісторію інтелектуалізації і, зокрема, теорії штучного інтелекту за певними періодами можна відобразити у таблиці 1.1 (символ \rightarrow означає «по теперішній час»).

Упродовж ХХ ст. теорія систем, системний аналіз, теорія штучного інтелекту одержали розвиток під впливом досягнень як класичних галузей науки (математика, фізика, хімія, біологія, історія й ін.), так і неklasичних областей (сінергетика, інформатика, когнітологія, теорія нелінійної динаміки й динамічного хаосу, катастроф, нейроінформатика, нейроінформатика й ін.), а також сучасного розвитку техніки і технологій. Ці досягнення пов'язані з такими іменами, як А. Тюрінг, Н. Вінер, Е. Ешбі, А.А. Ляпунов, Р. Акофф, І. Блауберг, Л. Бріллюен, Т.

Бусленко, Д. Гвішіані, Дж. ван Гіг, В.Т. Глушков, Т. Тесарович, Д. Поспелов, С. Янг, Г. Саймон, С. Тор-тон, Р. Спраг, Д. Пауер і чимало інших.

Таблиця 1.1 – Передісторія інтелектуалізації та штучного інтелекту

Період розвитку		Наука	Напрямки
з	по		
428 р. до н.е.	→	Філософія	Логіка, методи міркування, розум як фізична система на принципах навчання, мова, раціональність
800 р.	→	Математика	Формальне представлення, алгоритми, обчислення, розв'язність задач, теорія імовірності
1776 р.	→	Економіка	Практичні цілі, теорія рішень
1861 р.	→	Неврологія	Фізична основа розумової діяльності
1879 р.	→	Психологія	Явища сприйняття і моторний контроль, експериментальні техніки
1940 р.	→	Обчислювальна техніка	Високошвидкісні комп'ютери
1948 р.	→	Теорія управління і кібернетика	Проектування систем, що максимізують мету функціонування
1957 р.		Лінгвістика	Представлення знання, граматики

Сучасна історія штучного інтелекту за певними періодами відображена у таблиці 1.2 (символ → означає «по теперішній час»).

Таблиця 1.2 – Сучасна історія штучного інтелекту

Період розвитку		Події і досягнення
з	по	
1943 р.	1955 р.	Передумови штучного інтелекту
1956 р.		Народження штучного інтелекту (Dartmouth meeting, термін "Artificial Intelligence")
1952 р.	1969 р.	Ранній ентузіазм, великі очікування
1966 р.	1973 р.	Зіткнення з реальністю (штучний інтелект виявляє обчислювальну складність, дослідження нейронних мереж майже зникають)
1969 р.	1979 р.	Системи, ґрунтовані на знаннях, сподівання, що вони можуть стати ключем до успіху
1980 р.	→	Перетворення штучного інтелекту на індустрію
1986 р.	→	Повернення до нейронних мереж
1987 р.	→	Перетворення штучного інтелекту на науку
1995 р.	→	Поява підходу, ґрунтованого на використанні інтелектуальних агентів
2003 р.	→	Штучний інтелект рівня людини знову на порядку денному

«Передумови штучного інтелекту пов'язані з роботами МакКалока і Пітса (McCulloch & Pitts) щодо логічної циклічної моделі мозку (1943 р.), а також праця Тюрінга (Turing) «Computing Machinery and Intelligence» (1950 р.).

Ранній ентузіазм і великі очікування викликали перші дослідження нейронних мереж, виконані Мак-Калокком (1956 р.) та Фр. Розенблаттом (1962р.). Розенблатт створив нейронну мережу, яка моделювала роботу системи людського

зору – «око-зоровий центр мозку» – і отримала назву перцептрон (від латинського слова «perceptio» – сприйняття).

Суттєвий внесок в становлення нової науки внесли такі вчені, як Дж. Таккарті, автор першої мови програмування для задач штучного інтелекту ЛІСП, яка була розроблена в Стенфорді під його керівництвом на початку 60-х років, та М. Мінський (Масачусетський технологічний інститут), автор ідеї та винахідник фрейму та фреймової моделі представлення знань.

У середині 60-х та 70-х років проводились інтенсивні пошуки моделей та алгоритмів, здатних розв'язувати інтелектуальні творчі задачі. В 60-і роки народилась ідея лабіринтного пошуку, з'явилися перші програми для гри в шахи та шашки, алгоритм Робінсона (Robinson) для логічного міркування. 60-70-і роки – це епоха розвитку евристичного програмування. Родоначальниками цього цікавого напрямку були вчені А. Ньюелл (A. Newell), Г. Саймон та Дж. Шоу (John Clifford Shaw), які розробили систему доказу теорем Logic Theorist та систему GPS (General Problem Solver) – загальний розв'язувач задач. Ця система була з успіхом застосована для пошуку доведень теорем з евклідової геометрії на основі системи аксіом. На початку 70-х років з'явилася ще одна нова мова, здатна скласти конкуренцію ЛІСПу при реалізації систем, орієнтованих на знання, – мова ПРОЛОГ (розроблена в Марсельському університеті в 1971 р.).

Приблизно в цей же час суттєвий прорив в галузі ШІ було зроблено, коли на зміну пошуку універсальних алгоритмів мислення виникла ідея моделювати конкретні знання фахівців-експертів. В США з'явилися перші експертні системи, що базуються на знаннях – MYCIN (1976) та DENDRAL (1978), в результаті чого виник та почав застосовуватися новий підхід до розв'язання задач ШІ, заснований на представленні знань).

Окремі напрямки ШІ розвивалися в Японії, де в результаті було створено перший нейрокомп'ютер та перші інтелектуальні роботи.

Ці дослідження невдовзі привели до низки ефектних результатів. 1991 року під час Війни затоки (Gulf War) військові США розгортають систему логістичного планування та розкладу на основі методів штучного інтелекту, що контролювала до 50,000 транспортних засобів, вантажів та людей.

1997-го комп'ютер Deep Blue завдав поразки пануючому на той час шаховому чемпіонові світу Гаррі Каспарову. Система Proverb розв'язує головоломки кросворду краще, ніж більшість людей. У 2016-му програма AlphaGo виграє матч у Лі Седоля, одного з найсильніших професійних гравців світу в гру «Го». З застосуванням програми доведено математичну здогадку (здогадка Робінсона), що була невирішеною впродовж десятиліть.

«Без рук через Америку» (Look, Ma, no hands!) – транспортний засіб проїхав, управляючись автономно 98 % маршруту від Пітсбургу до Сан Дієго. Бортова автономна плануюча програма NASA управляє діями космічного літального апарату.

В Україні наукові дослідження у сфері штучного інтелекту сконцентрувалися в Інституті кібернетики НАНУ. У 1970-80 рр. проводяться роботи щодо нових методів, алгоритмів розпізнавання зображень (професори В.А. Ковалевський, Т.І. Шлезінгер, В.І. Рибак), розпізнавання мовних сигналів (професор Т.К. Вінцюк), навчання та самонавчання розпізнаючих систем (академік НАНУ О.Г. Івахненко та його учні). Праці О.Г. Івахненко також присвячені створенню принципово нового методу індуктивного моделювання складних систем – метод групового урахування аргументів (МГУА). Проблемами моделювання мислення та психіки займався академік М.М. Амосов. Внесок в галузь нейронних мереж та нейрокомп'ютерів зробили Е.М. Куссуль, О.М. Різник (Інститут проблем математичних машин і систем НАН України) та інші. Також у цій царині необхідно відзначити внесок докторів наук Л.С. Ямпольського (НТУУ КПІ), Е.В. Бодянського (ХНУРЕ). На розробку інтелектуальних систем планування дій були направлені дослідження професора В.П. Гладуна.

В наші часи дослідження і розвитку СППР проводяться в Інституті математичних машин і систем під керуванням члена-кореспондента НАНУ А.О. Морозова. Розвідки щодо моделювання соціальних об'єктів проводяться в Інституті прикладного системного аналізу (ІПСА) НТУУ КПІ (академік Згуровський М.З., доктори наук Панкратова Н.Л., Бідюк П.І, Зайченко Ю.П.).

ТЕМА 2 ПІДХОДИ ДО ІНТЕЛЕКТУАЛЬНИХ СИСТЕМ

2.1 Основні підходи

З попередньої теми витікає важливий висновок, що розвиток інтелектуальних систем тісно пов'язаний з роботами та дослідженнями у напрямку штучного інтелекту, спробами різними шляхами реалізувати аналог мислення людини.

Нині при побудові інтелектуальних систем (ІнС) і виборі методу представлення знань використовуються різні підходи або їх комбінації, а саме:

- структурний;
- логічний;
- еволюційний;
- імітаційний.

Структурний підхід. Назва структурного підходу пов'язана із спробами побудови ІнС шляхом моделювання на комп'ютері структури людського мозку, що

включає моделі нейронів мозку і нейронних мереж. Однією з перших таких реалізацій структурного підходу став простий перцептрон Розенблатта, запропонований ним і досліджений у 1957-1959 рр. для розпізнавання зорових образів. Пізніше цей напрям розвинувся в теорію розпізнавання образів і штучні нейронні мережі, і вважається одним з перспективніших напрямів створення ІНС.

Однак тут є чимало проблем. Відповідно до сучасних уявлень мозок людини складається приблизно з 10 мільярдів нервових клітин – нейронів. Для моделювання (навіть не дуже точного) одного нейрона потрібні досить складні електронні схеми, а для прояву більш-менш цікавих для практики інтелектуальних властивостей потрібно створювати ансамблі з вельми значної кількості нейронів. Вченими японських і німецьких наукових центрів була створена модельна мережа, що складається з 1,73 млрд. нервових клітин і 10,4 трлн синапсів (засобів з'єднання клітин). Імітація однієї секунди активності навіть на такій моделі, приблизно еквівалентній 1 % реального мозку людини, потребувала 40 хвилин машинного часу одного з самих швидкодіючих суперкомп'ютерів! .

Крім того, структурування моделі мозку до рівня нейронів, як виявляється, ще не гарантує успіху. Так, наприклад, Говард Гарднер – американський психолог, автор теорії множинного інтелекту, що стала класичною, вважає, що людина має не єдиний інтелект (так званий «загальний інтелект»), а низку відносно незалежних здібностей! Серед останніх, на думку автора, критеріям інтелекту відповідають лінгвістичний, музичний, логіко-математичний, просторовий, тілесно-кінестетичний, внутрішньо-особовий і між-особовий інтелекти. Моделювання цих інтелектуальних можливостей, що взаємодіють, на структурному рівні є задачею вельми нелегкою.

Логічний підхід. Цей підхід є найбільш поширеним. Його виникнення пов'язане безпосередньо із здібностями людини до розвиненого логічного мислення, що відрізняє його принципово від тварин. Основи логічного підходу почали розвиватися від логіки Аристотеля, числення висловлювань і булевої алгебри. Фактично усі побудовані на логічному принципі ІНС реалізують так звані «машини доказу теорем». Кожна така машина має блок генерації мети і систему висновку, що намагається довести цю мету як теорему. Класичний логічний підхід для більшості реальних задач характеризується великою трудомісткістю з-за можливого повного перебору варіантів під час пошуку доказу.

Еволюційний підхід використовується також досить широко при побудові ІНС. У цьому підході основна увага приділяється побудові початкової моделі і правилам, за якими вона може змінюватися (еволюціонувати). При цьому модель може бути складеною з використанням найрізноманітніших методів, включаючи нейронні мережі, набори логічних правил і будь-які інші моделі. Окрім правил в

програмі визначаються також критерії оцінки якості кожного варіанту. Нині цей напрям отримав розвиток під назвою «генетичні алгоритми». Це алгоритм, що оперує з популяцією індивідів $P(t) = \{x_i(t), \dots, x_l(t)\}$, де $t = 1, 2, \dots$ номер ітерації. Розроблено чимало різних модифікацій генетичних алгоритмів, що утворюють цілий клас методів. До цього класу можна віднести і так званий метод групового урахування аргументів (МГУА), який активно пропагувався в нашій країні академіком А.Г. Івахненком ще у 80-х рр. минулого століття. Сучасні вдосконалені робочі варіанти генетичних алгоритмів дозволяють у ряді випадків отримувати досить ефективні результати при розумному обмеженні множини допустимих наборів варійованих операторів.

Підхід, що ґрунтується на введеному американським кібернетиком У.Р. Ешбі класичному базовому понятті кібернетики «чорного ящика» є імітаційними. Модель чорного ящика представлена на рисунку 2.1.

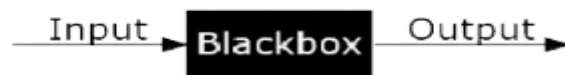


Рисунок 2.1 – Модель чорного ящика

Модель об'єкту дослідження будується на основі його поведінки, реакцій на дії, що надходять ззовні на його входи, характеризує зв'язки між реакціями і діями, що викликали їх, і зовні імітує здатність людини копіювати поведінку інших, не розуміючи, як це відбувається і чому. Основним недоліком імітаційного підходу (як і еволюційного) є низька інформаційна здатність більшості побудованих моделей (у сенсі розуміння структури і параметрів моделі «внутрішнього» устрою чорного ящика).

Необхідно зазначити, що при побудові систем штучного інтелекту для управління в технічних системах класична модель чорного ящика використовується дуже рідко. Про реальні технічні об'єкти і системи зазвичай все ж є та або інша апріорна і поточна інформація. Тому в застосуваннях ШІ до технічних систем частіше розглядають так звані «сірі», або навіть «білі» ящики. Наявність тих або інших знань і даних про сам об'єкт ще не гарантує можливість гарного управління об'єктом. По-перше, будь-яке знання і будь-яка модель завжди лише приблизно відображають реальний об'єкт. По-друге, потрібні знання про збурення, що можливо діють на об'єкт і, нарешті, необхідно мати або уміти формувати цілі інтелектуальної системи для різних рівнів управління.

Комбінаційний підхід. На верхніх рівнях управління для прийняття рішень частіше використовуються логічні підходи. Завданнями систем нижніх рівнів

управління є забезпечення їх взаємодії із зовнішнім середовищем отримання і первинна обробка інформації і формування реакцій системи залежно від рішень, прийнятих на верхніх рівнях. Тому в системах нижнього рівня, де відбувається обробка первинної інформації, частіше використовується комбінація структурного, еволюційного і імітаційного підходів.

Природно, вказані підходи виникли не раптово, а поступово розвиваючись на основі певних теоретичних гіпотез. При цьому можна виділити певні етапи становлення, а саме: евристичний пошук, експертні системи, гібридні інтелектуальні системи, включаючи системи м'яких і інтелектуальних обчислень, штучні агенти і багатоагентні системи, онтології, інтелектуальні середовища. Що ховається за цими етапами?

Як вказувалося, перші дослідження можливостей інтелектуалізації були пов'язані з аналізом ігор, які можуть породжувати надзвичайну кількість просторів станів. Для пошуку в них потрібні потужні методики, що визначають, які альтернативи слід розглядати. Такі методики називають евристиками і складають значну область досліджень ШІ. Прикладом евристики може бути рекомендація перевіряти, чи включений прилад в розетку, перш ніж робити припущення про його поломку, або виконувати рокіровку в шаховій грі, щоб спробувати уберегти короля від шаху. Велика частина того, що ми називаємо розумністю, мабуть, спирається на евристики, які люди використовують у вирішенні завдань.

Евристика – стратегія корисна, але потенційно здатна упустити правильне рішення. Тому зусилля були направлені на спроби накопичення в комп'ютері знань людей-фахівців з певної області, які потім можна використовувати для прийняття рішень, що привело до створення експертних систем.

Але ці системи також не позбавлені недоліків, тому останніми роками інтенсивно розвиваються гібридні інтелектуальні системи, що дозволяють використати переваги традиційних засобів і методів штучного інтелекту, і, в той же час, подолати деякі їх недоліки. Прикладом гібридних ІНС служать гібридні експертні системи, що є інтеграцією експертних систем і нейронних мереж. Вони поєднують як знання, що формалізуються (в базах знань експертних систем), так і знання, що не формалізуються (у нейронних мережах). Гібридні інтелектуальні системи працюють з невизначеністю, об'єднуючи такі області як нечітка логіка, нейронні мережі, імовірнісні міркування, еволюційні алгоритми та ін., які доповнюють один одного і використовуються в різних комбінаціях. Для оперування ними фактично було створено новий метод обчислювальної математики, який отримав назву «м'які обчислення». Цей термін був уведений американським вченим Лофті Заде у 1994 році.

Подальші намагання підсилити підтримку осіб, що приймають рішення, привели до створення методів і інструментальних засобів, що отримали назву інтелектуальних обчислень. До них, у першу чергу, відносяться обчислення, що базуються на онтологічному та когнітивному моделюванні. Онтології були запропоновані Т. Грубером для декларативного подання знань і визначаються як бази знань спеціального виду, як «специфікація концептуалізації» предметної області. Когнітивне моделювання, першопрохідцем якого був Ван Хао, використовує подання у вигляді орієнтованих графів, що описують причинно-наслідкові зв'язки об'єктів і понять предметної області.

Подальші розвідки привели до того, що погляди різних вчених на штучний інтелект стали усе більше відрізнятися, породжуючи різні підходи до проблеми. Одним з таких був прийнятий підхід, що знайшов чимало прихильників, згідно з яким інтелектуальність в основному пов'язана з раціональною діяльністю. В рамках цього підходу сформувався поняття агента і багатоагентних систем.

Агентом вважається все, що діє (слово агент пішло від з латинського слова *agere* – діяти). Функція агента для штучного агента реалізується за допомогою програми агента, але вважається, що інтелектуальні агенти мають деякі інші атрибути, які відрізняють їх від звичайних «програм», такі як здатність функціонувати під автономним управлінням, сприймати своє середовище, адаптуватися до змін і мати здатність узяти на себе досягнення цілей, поставлених іншими. Агент, який діє так, щоб можна було досягти найкращого результату (або найкращого очікуваного результату), отримав назву раціонального агента.

За визначенням В.К. Фінна (радянського вченого) агента (або інтелектуального робота) можна розглядати як когнітивну систему, що має можливість дії після прийняття рішення. Тоді загальна структура інтелектуального агента відповідає наступній схемі, що відображена на рисунку 2.2:

інтелектуальний агент = підсистема сприйняття + інтелектуальна система + підсистема дії.

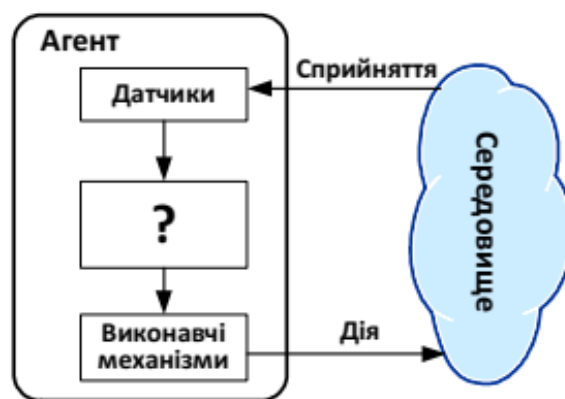


Рисунок 2.2 – Взаємодія агента з середовищем

З цього випливає визначення для інтелектуального середовища: інтелектуальне середовище = розподілена система сприйняття параметрів фізичного середовища + інтелектуальне ядро + розподілена система дії на фізичне середовище.

Розподіленість інтелектуального середовища реалізується сукупністю агентів (багатоагентністю). Багатоагентна система (Multi-agent system – МАС) – це система, утворена декількома взаємодіючими інтелектуальними агентами. Багатоагентні системи є перспективною технологією на принципах самоорганізації та самонавчання. Вони виявилися новою концептуальною парадигмою для аналізу задач та синтезу систем, багатообіцяючим підходом до подолання проблем, пов'язаних зі складністю, розподіленістю та інтерактивністю.

Багатоагентні системи вже зараз успішно використовуються у різних областях діяльності. Зазвичай в них під агентом розуміється мобільний робот (mobile robot), безпілотний літальний апарат (*unmanned flying vehicle*) або автономний підводний апарат (*autonomous underwater vehicle*), що в основному займаються збиранням інформації про середовище, що оточує.

2.2 Системи, що ґрунтуються на знаннях

Знання на сьогодні є найважливішим активом сучасного підприємства (організації). Знання – це інформація в контексті, здатна спонукати до дій розуміння. Розділяють дві основні категорії знання – явне і приховане. Явне знання може бути описаним на формальній мові і передане іншим людям, зокрема й в пам'ять автоматизованої системи, а приховане може бути охарактеризоване як персональне знання, пов'язане з особистим досвідом і такими невідчутними чинниками, як особисті переконання, перспективи і цінності.

Для чого потрібні знання в інтелектуальних системах? Передусім, системи на знаннях є перспективними для реалізації верхніх (високоінтелектуальних) рівнів управління, там, де приймаються рішення, як показано на рисунку 2.3.

Адже в сучасних умовах складність проблем, що вирішуються в сфері управління постійно зростає, а зв'язки між причинами і наслідками усе далі відсуваються від поняття порядку в сферу зростання складності і навіть хаосу.



Рисунок 2.3 – Використання знань в інтелектуальних системах

Зв'язки між причинами та наслідками в сфері прийняття рішень відображено на рисунку 2.4.



Рисунок 2.4 – Зв'язки між причинами та наслідками в сфері прийняття рішень

Внаслідок цього для прийняття рішень потрібно усе більше знань. У процесі прийняття рішення особа використовує реальні знання, тобто ті, що є в неї в наявності.

Але знання, потрібні для вирішення проблеми, можуть знаходитись й в інших станах, які поазано на рисунку 2.5. Вони (стани) можуть представляти так звані «сліпі плями», коли особа прийняття рішень (ОПР) знає, що в неї не вистачає деяких знань для розв'язання задачі. Найгірший варіант, коли ОПР й не знає, що десь є знання, які їй є такими необхідними (невідомі «сліпі плями»). Образливо, коли потрібні знання є, але особа про це й не знає. Ефективна інтелектуальна система

саме й призначена для того, щоб допомогти особі-користувачу подолати ці перепони.



Рисунок 2.5 – Стани знання в особі, що приймає рішення

Для того, щоб користуватися знаннями, необхідно навчити комп'ютерні системи маніпулювати ними. В рамках цього напрямку створюються засоби отримання (добування) знань від фахівців-експертів і поповнення баз знань, вивчаються системи класифікації знань, розробляються процедури узагальнення знань і формування на їх основі понять. Також створюються методи правдоподібного виведення нових знань на основі наявних з використанням різних правил висновку. Даний напрямок глибоко пов'язаний, у першу чергу, із створенням, експертних систем.

Системи, що ґрунтуються на знаннях, можливо класифікувати наступним чином:

- системи, що ґрунтуються на правилах;
- системи, що ґрунтуються на методах автоматичного доказу теорем;
- системи, що ґрунтуються на автоматичному гіпотезуванні;
- системи, що ґрунтуються на міркуваннях по аналогії;
- об'єктно-орієнтовані системи.

Найпоширенішими є системи, що ґрунтуються на правилах, які поділяються на:

- системи на базі продукційних правил (типу «якщо виконується умова А, тоді роби В, інакше роби С»);

- системи на нечітких правилах, що формулюються в зручних для людини якісних термінах, а саме в термінах нечітких понять (багато, мало, добре, погано, невідомо, і т.ін.);
- системи логічного програмування.

Для визначення формальних систем важливим є поняття логічного висновку. Для цього необхідно задати:

- 1) мову системи, в якій визначається алфавіт мови як деяка множина символів і правил побудови з символів цього алфавіту елементів мови, що називаються формулами визначуваної мови;
- 2) аксіоми системи, що виділяються як деякі з формул мови формальної системи;
- 3) правила висновку системи.

2.3 Напрямки застосування засобів штучного інтелекту в інтелектуальних системах

Напрямки застосування засобів ШІ в інтелектуальних системах визначаються задачами моделювання основних функцій людського інтелекту, рисунок 2.6.



Рисунок 2.6 – Задачі моделювання функцій людського інтелекту

Розпізнавання образів. Задача розпізнавання образів відноситься до задач класифікації і передбачає складання описів об'єктів і правил, що визначають по цих описах приналежність об'єктів до тих або інших класів. Поняття образу в цьому визначенні еквівалентно поняттю класу.

При розпізнаванні зорових образів в разі простих (не складених) образів найбільше застосування отримує метод порівняння з еталонами. На практиці найчастіше застосовується прийом, коли заздалегідь можна виділити контури

спостережуваного об'єкту і здійснювати порівняння також з контурною проекцією еталону.

Іншим важливим напрямком є розпізнавання *і генерація мовної інформації*. Ця задача має два боки – фізичний і смисловий. З одного боку, подібно до того, як письмова мова представляється послідовністю букв, усна мова може бути представлена у вигляді послідовності елементарних звуків, що отримали назву фонем, які оцифровуються і записуються у пам'ять комп'ютера. Задача автоматичного розпізнавання мови незрівнянно складніша за задачу її автоматичної генерації (синтезу) і розв'язується поки що на рівні, що поступається можливостям людського слуху і мозку.

Планування цілеспрямованих дій. Важливою для інтелектуальних систем є й задача моделювання функції людського інтелекту, що полягає у плануванні цілеспрямованих дій. Вона може виникати як в статичному вигляді, коли мета в процесі її досягнення не змінюється, так і в динамічному, коли мета змінюється. Таку задачу інколи називають задачею цілеполагання. Розв'язання задачі зводиться до пошуку найкоротшого шляху на графі, що описує шляхи до мети. Зазвичай використовуються алгоритми пошуку у ширину, що простіше, або у довжину графа, що більш притаманне мисленню людини, послідовним дослідженням вершин графа. В складних задачах планування цілеспрямованої поведінки великого значення набувають різного роду евристичні прийоми (зазвичай сформовані в результаті набутого досвіду) постановки досить крупних проміжних цілей і використання методів досягнення цілей, що застосовуються не до кінцевих, а саме до цих проміжних цілей. Саме так поступає, наприклад, шахіст, який шукає шляхи до виграшу партії, вирішуючи проблеми конкретної ситуації на шахівниці у певній стадії гри.

Для інтелектуальної підтримки прийняття рішень часто є корисним моделювання ізрових задач. В цих задачах процес проведення гри – це процес послідовного розгортання учасниками гри деякого шляху на графі можливих станів гри. Оскільки глибина дерева (графу) гри зазвичай є великою, то повне перебирання можливих шляхів вимагає занадто багато часу. У зв'язку із цим використовують так звані оцінні функції, що оцінюють аналізовані завершальні (для даного рівня перебору) позиції і вибирають черговий хід так, щоб при будь-яких можливих продовженнях мінімальне значення цільової функції («проміжного виграшу» гравця, що робить хід) було б максимальним.

Зрозуміло, що розв'язання перерахованих задач неможливе без застосування відповідних математичних методів, які були сформовані у процесі досліджень питань моделювання штучного інтелекту. В ІнС знайшли застосування такі сьгоднішні методи, як, перед усім, методи нейронних мереж, методи евристичної

самоорганізації, генетичні алгоритми, а також методи байєсових мереж, моделі марковського типу, зокрема приховані марковські моделі, метод групового урахування аргументів, лінгвістичне моделювання та ін.

ЗАВДАННЯ ДЛЯ ПЕРЕВІРКИ ЗНАНЬ

1. Що на сьогодні є найважливішою складовою процесів прийняття рішення?
2. Що таке інтелектуалізація інформаційних систем?
3. У чому полягають переваги інтелектуальних систем?
4. Які напрямки штучного інтелекту застосовуються для інтелектуалізації інформаційних систем?
5. Що на сьогодні можна віднести до основних інтелектуальних технологій?
6. Які підходи нині використовуються при побудові інтелектуальних систем?
7. У чому полягають переваги гібридних інтелектуальних систем?
8. Які атрибути відрізняють інтелектуальні агенти від звичайних програм?
9. Які задачі моделювання основних функцій людського інтелекту визначають напрямки застосування засобів штучного інтелекту в інтелектуальних системах?
10. У чому користь моделювання ігрових задач для інтелектуальної підтримки прийняття рішень?

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Нестеренко О. В., Ковтунець О. В., Фаловський О. О. Інтелектуальні системи і технології. Ввідний курс: навч. посібник. / за ред. О. В. Нестеренко. Київ: Національна академія управління, 2023.
2. Величко О.М., Гордієнко Т.Б. Інтелектуальні інформаційні системи: структура і застосування: підручник. Одеса: Олді+, 2022. 728 с.

РОБОТОТЕХНІЧНІ ТА ІНТЕЛЕКТУАЛЬНІ МЕХАТРОННІ ПРИСТРОЇ (укладач Федік Л. Ю.)

ТЕМА 1 ХАРАКТЕРИСТИКА РОБОТІВ І МАНІПУЛЯТОРІВ

1.1 Класифікація роботів

Робот - це автоматичний пристрій призначений для виконання виробничих і інших операцій, які зазвичай виконуються людиною (рис. 1.1).



Рисунок 1.1 – Промислові роботи кука в автомобільному виробництві

Загальна класифікація промислових роботів містить від 7 до 12 класифікаційних ознак (у залежності від інформаційного джерела). Сфера використання роботів прогресивно розширюється і відповідно зростає перелік типів роботів, які визначаються основним їх призначенням.

Розглянемо класифікацію роботів за основними ознаками. Першою ознакою, є їх призначення, тобто галузь застосування. Згідно неї роботи розділяють на *промислові* і *сервісні*.

Сьогодні основним таким типом, як і раніше, є промислові роботи (пр), які призначені для застосування в промисловості і складають до 80% всього парку роботів у світі. У свою чергу промислові роботи діляться на типи вузького призначення (наприклад, робот фарбувальний, зварювальний, транспортний, для обслуговування верстатів, пресів, ливарних машин і т.д.).

За типом виконуваних операцій всі промислові роботи діляться на роботи *технологічні*, які виконують основні технологічні операції, і роботи *допоміжні*, що виконують допоміжні технологічні операції з обслуговування технологічного

обладнання. Технологічні роботи відносяться до основного технологічного устаткування, а допоміжні можна віднести до засобів автоматизації.

За операціями, для виконання яких призначений робот, розрізняють роботи *спеціальні, спеціалізовані й універсальні*. *Спеціальні* роботи призначені для виконання однієї конкретної технологічної операції (наприклад, обслуговування певної марки технологічного обладнання). *Спеціалізовані* роботи можуть виконувати різні однотипні операції (складальний робот із змінними робочими інструментами, робот для обслуговування певного типу технологічного устаткування і т.п.). *Універсальні* роботи призначені для виконання будь-яких основних і допоміжних операцій в межах їх технічних можливостей.

Промисловий робот складається з *маніпулятора*, в тому числі приводів, і контролера, включаючи пульт навчання і апаратний або програмний комунікаційний інтерфейс. Маніпулятор робота можна програмувати за трьома або більше ступенів рухливості в залежності від завдань автоматизації.

Класифікація роботів за способом керування: роботи з *програмним, адаптивним і інтелектуальним* керуванням. Керування рухом за окремими ступенями рухливості може бути *безперервним* (контурним) і *дискретним* позиційним.

В останньому випадку керування рухом здійснюють, задаючи кінцеву послідовність точок (позицій) і подальше переміщення по них кроками від точки до точки без контролю траєкторій між цими точками. Найпростішим варіантом дискретного керування є *циклове*, за якого кількість точок позиціонування по кожній ступені рухливості мінімальне, тобто найчастіше обмежена двома - *початковою і кінцевою*.

1.2 Технічні характеристики роботів

До показників, що визначають технічні характеристики роботів відносять: *тип приводів* робота, його *вантажопідйомність*, *кількість маніпуляторів*, *тип і параметри* їх робочої зони, *рухливість* і *спосіб розміщення*, виконання за *призначенням*.

Приводи, які використовуються в маніпуляторах і системах пересування роботів, діляться на *електричні, гідравлічні і пневматичні*. Часто їх застосовують в комбінації.

Вантажопідйомність робота - це вантажопідйомність його маніпуляторів, а для транспортного робота ще і його шасі. Вантажопідйомність маніпулятора визначається масою переміщуваних ним об'єктів і в залежності від призначення робота може становити від одиниць грама (надлегкі роботи, наприклад, що

застосовуються в мікроелектроніці) до декількох тисяч кілограм (надважкі, наприклад, транспортні та космічні роботи).

Кількість маніпуляторів у роботів у більшості випадків обмежена тільки одним (*одноманіпуляторні* або *однорукі* роботи). Однак в залежності від призначення, існують конструкції роботів із *двома, трьома і чотирма* маніпуляторами. Наприклад, існують промислові роботи для обслуговування пресів холодного штампування з двома різними маніпуляторами: один основний для взяття заготовки і установки її в прес, а інший спрощеної конструкції для виконання більш простої операції зштовхування готової деталі в бункер.

Тип і параметри робочої зони маніпуляторів робота визначають область навколишнього його простору, в межах якої робот може здійснювати маніпуляції.

Робоча зона маніпулятора - це простір, в якому може перебувати його робочий орган за всіх можливих пересувань.

Рухливість робота визначається наявністю чи відсутністю у 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 і гусеничних* до призначених для пересувань у воді, глибинах землі, в повітрі і космосі. Специфічним для робототехніки способом пересування є *крокування*.

За способом *розміщення* стаціонарні і мобільні роботи бувають *підлоговими, підвісними* (мобільні роботи цього типу зазвичай переміщуються по рейковому шляху) і т. Д.

Виконання робота за призначенням залежить від зовнішніх умов, у яких він повинен функціонувати. Розрізняють виконання *нормальне, пилозахисне, теплозахисне, вологозахисне, вибухобезпечне* і т. д.

1.3 Основні параметри роботів

До основних параметрів роботів, що визначають їх динамічні властивості відносяться *швидкодія і точність* їх рухів. Ці параметри взаємопов'язані й складаються з їх значень для маніпуляторів і систем пересування.

Швидкодія маніпулятора визначається швидкістю *переміщення* робочого органу.

Швидкодію маніпуляторів у роботів загального застосування можна розбити на *три* діапазони:

1. Малу - за лінійних швидкостей до 0,5 м/с;
2. Середню - за лінійних швидкостей від 0,5 до 1-3 м/с;
3. Високу - за великих швидкостей.

Найбільша швидкість маніпуляторів сучасних роботів досягає 10 м/с і вище.

Для значної частини сфер застосування роботів швидкодія є дуже важливим параметром, оскільки визначає їх *продуктивність*. Основна складність при підвищенні швидкодії пов'язана з відомим протиріччям між швидкодією і *точністю*.

Точність маніпулятора і системи пересування робота характеризується результируючою похибкою позиціонування (за дискретного руху) або відпрацювання заданої траєкторії (за безперервного руху).

Найчастіше точність роботів характеризують *абсолютною похибкою*. Точність роботів загального застосування можна розбити на три діапазони:

1. Малу - за лінійної похибки від 1мм і більше;
2. Середню - за похибки від 0,1 до 1 мм;
3. Високу - за меншої лінійної похибки.

Найменшу точність мають роботи призначені для виконання найбільш грубих, наприклад, транспортних рухів, а найбільшу мікронну - роботи, які використовуються в електронній промисловості.

Поряд із класифікаційними параметрами, роботи характеризуються параметрами, які визначають їх *технічний рівень*. До них відносяться і деякі з розглянутих вище параметрів, які можуть мати кількісне вираження, як то швидкодія й точність.

Іншими параметрами, котрі характеризують технічний рівень роботів, є *надійність*, *кількість* одночасно працюючих *ступенів рухливості*, *час програмування*, а також засновані на перерахованих вище параметрах різні *відносні* і *комбіновані* показники.

До них відносяться, зокрема, *питома вантажопідйомність*, віднесена до маси робота, *вихідна потужність* маніпулятора, яку відносять до потужності його приводів, відносні оцінки *габаритних* параметрів, маніпуляційних *кінематичних* і *динамічних* характеристик, *можливостей програмування*, *економічної ефективності* і т.п.

Однак ці відносні показники технічного рівня вже не є паспортними параметрами, які використовуються для характеристики конкретних роботів, а слугують *критеріями якості*, призначеними для їх оптимізації під час проектування і порівняльної оцінки.

1.4 Будова промислових роботів

Приведемо більш розширений варіант визначення промислового робота та його складових блоків.

Промисловий робот - це автоматична машина, стаціонарна чи пересувна, яка складається з виконавчого пристрою у вигляді *маніпулятора*, що має кілька

ступенів рухливості, і пристрою програмного керування для виконання рухових і керуючих функцій у виробничому процесі.

Маніпулятор – це керований пристрій призначений для виконання рухових функцій під час переміщення об'єктів у просторі, оснащених робочим органом. Він складається з несучих конструкцій, виконавчих механізмів, пристрою захвату, приводу з передавальними механізмами і пристрої пересування. Несучі конструкції слугують для розміщення всіх пристроїв і агрегатів робота, а також для забезпечення необхідної міцності і жорсткості маніпулятора. Вони виконуються у вигляді підстав, корпусів, стоек, рам, візків, порталів і інших.

Виконавчий механізм – це сукупність рухливо з'єднаних ланок маніпулятора, призначених для впливу на об'єкт маніпулювання або обробляюче середовище.

Робочий орган – це складова частина виконавчого механізму маніпулятора для безпосереднього виконання технологічних операцій і (або) допоміжних переходів (наприклад, складальний інструмент, пристрій захвату).

Пристрій захвату – це кінцевий вузол маніпулятора, що забезпечує захоплення і утримання в певному положенні об'єкта маніпулювання. Привід призначений для перетворення енергії, що підводиться в механічний рух ланок виконавчого механізму відповідно до сигналів, які надходять від пристрою керування.

Пристрій пересування слугує для переміщення маніпулятора або промислового робота в цілому в потрібне місце робочого простору. Він складається з *ходової* частини і *приводних пристроїв*.

На рисунку 1.2 представлена одна з конструкцій промислового робота.

На рисунку 1.3 зображено спрощену блок-схему промислового робота. Промислові роботи можуть мати різну будову, що багато в чому визначається завданнями, які стоять перед ними.

На даний період розвитку робототехніки найбільш поширеними видами є *роботизовані маніпулятори*.

Стандартний маніпулятор виконаний з семи сегментів, які з'єднуються за допомогою шести суглобів. Кожен сегмент виконаний з металевого корпусу і проводів.

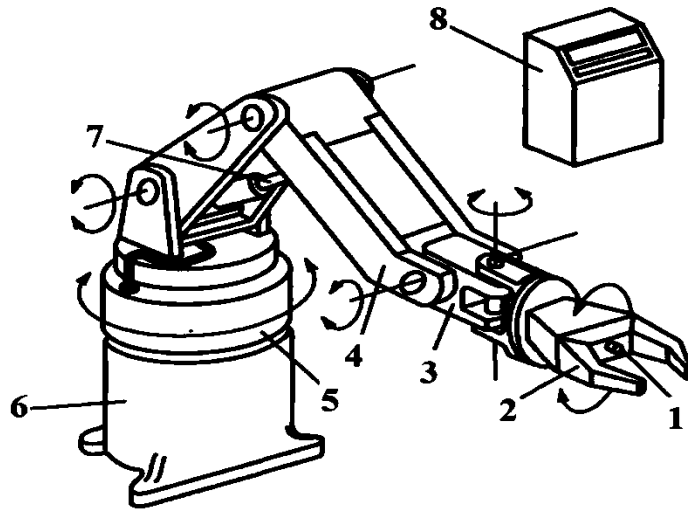


Рисунок 1.2 – Приклад конструкції промислового робота: 1 – датчик зворотного зв’язку; 2 – пристрій захвату; 3 – кість маніпулятора; 4 – рука маніпулятора; 5 – колона; 6 – несуча конструкція; 7 – привод руки маніпулятора; 8 - блок керування з пультом



Рисунок 1.3 – Спрощена блок-схема промислового робота

У кожному сегменті присутні крокові двигуни, які змушують суглоби рухатися.

Керування суглобами і роботом в цілому здійснюється за допомогою комп'ютера, який змушує обертатися конкретні крокові двигуни.

У деяких маніпуляторах замість двигунів можуть застосовуватися пневматичні або гідравлічні пристрої.

Головна особливість крокових двигунів в тому, що вони можуть забезпечувати дуже точні рухи. Якщо комп'ютер дасть команду суглобу пересунути рівно на 15 мм, то двигун виконає завдання. При цьому рух буде точним кожен раз при вчиненні однієї і тієї ж дії.

Щоб контролювати, чи правильно робот виконує необхідні дії, застосовуються *датчики руху*. Якщо відбувається найменше відхилення від заданої програми, то відбувається корекція руху. Якщо ж спостерігається значне відхилення і неможливість його виправлення, то сигнал подається на головний комп'ютер.

У результаті робот може бути зупинений, щоб його можна було відремонтувати.

Маніпулятор також має *пристрій захоплення*, який виконаний у вигляді людської руки з механічними пальцями. При необхідності захоплення плоского предмета замість механічних пальців може застосовуватися пневматична присоска. У разі необхідності захоплення безлічі деталей може бути задіяна конструкція у вигляді пристосування, спеціально розробленого для цього. До того ж замість пристроїв захвату можуть застосовуватися і інші робочі інструменти, наприклад, *пультверизатор*, *викрутка* і так далі.

Промислові роботи також можуть переміщатися по колії, виконаній на підлозі у вигляді монорейок. У разі необхідності руху по нерівній поверхні, використовуються інші конструкції, наприклад, пневматичні присоски тощо. Для живлення роботів може використовуватися *акумулятор*, але найчастіше застосовується промислова електрична мережа. Для керування використовується комп'ютер, який командує роботом через дроти або бездротовий зв'язок. Також в самих роботах може бути встановлений *блок пам'яті*, куди записується необхідна програма.

Промислові роботи, які мають шість суглобів, зовні нагадують людську руку (плече, лікоть і зап'ястя). Роботизовані руки за своєю будовою нагадують руку людини. У людей є лікоть і плече, суглоби, що дозволяють рукам вільно згинатися, а також кістки з'єднують суглоби. Такий принцип руху суглобів і передачі потужності через зв'язки типовий як для людей так і для роботів. На рис. 1.4 зображено фізичну модель руки. Рух суглобів - досить тонкий процес. Кабелі і дроти вбудовуються в руку, що дозволяє уникнути створення перешкод периферійного устаткування. У більшості випадків плече монтується на нерухомій основі. У результаті такої будови робот може мати шість ступенів свободи, а це значить, що він здатний рухатися за шістьма різними напрямками.

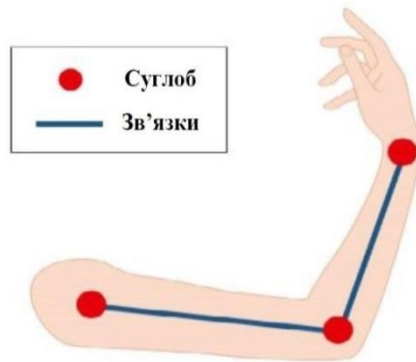


Рисунок 1.4 – Зображення фізичної моделі руки

Подібно людській руці маніпулятор також переміщує кінцевий ефектор з одного місця на інше. При оснащенні кінцевого ефектора різними пристроями, у робота з'являється можливість виконувати певні технологічні операції. Одним із найбільш поширених варіантів є подоба руки, що дозволяє роботу захоплювати і переміщувати об'єкти з місця на місце (рис. 1.5).

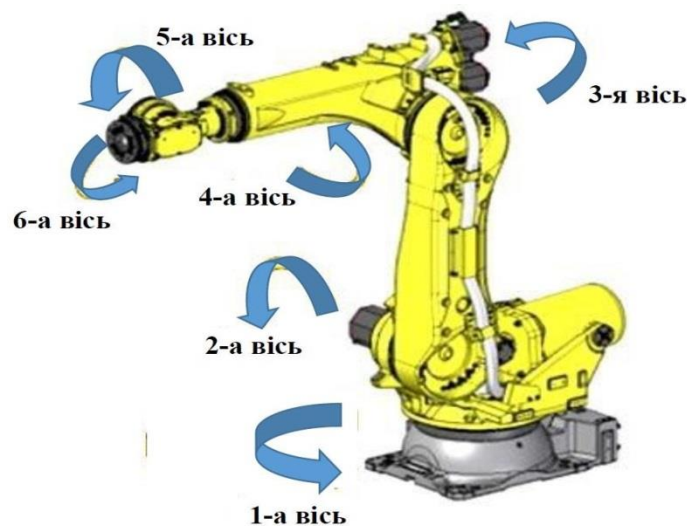


Рисунок 1.5 – Робот «людська» рука з механічними пальцями

Досить часто маніпулятори мають вбудовані датчики тиску, завдяки чому вони можуть контролювати силу захоплення. Якщо сила буде занадто велика, то, наприклад, лампочка, як об'єкт дії, просто лопне. Контролювання сили стиснення гарантує, що лампочка не постраждає (рис. 1.6).



Рисунок 1.6 – Робот «людська» рука з датчиком тиску

За допомогою інших кінцевих ефекторів можуть використовуватися розпилювачі порошку, різні дрилі і так далі.

Керування такими роботами може бути виконане: за допомогою програми, адаптивного керування, безпосередньо людьми, але на відстані, а також своєрідним «штучним інтелектом».

Під час експлуатації роботів використовуються сенсорні пристрої, сигнали яких аналізуються, після чого виконується необхідна дія.

Слід прийняти до уваги, що одні типи роботів призначені тільки для виконання певної технологічної операції, інші – можуть виконувати відразу кілька.

Промислові роботи - це технічно складні пристрої, які вимагають грамотного програмування, налаштування і обслуговування.

1.5 Система маневрування промислових роботів

У процесі експлуатації промислових роботів величезне значення має система маневрування, тобто *кінематична* система. Головне її завдання полягає в забезпеченні переміщення об'єкта – оброблюваної деталі чи інструменту - з однієї точки в іншу, тобто, на будь-яку позицію в рамках робочої зони. Для цього передавальний механізм направляючого грейфера має *ротаційні* і *трансляційні* вузли.

Трансляційні вузли слугують для виконання прямолінійних переміщень. Їх зв'язок між собою здійснюється за допомогою ротаційних вузлів, які забезпечують можливість роботи за рахунок дії приводів виконувати обертання і поворот руху. Комбінуючи ці вузли між собою, можна встановити межі робочої зони робота.

Простежимо це на прикладі. Тіло (припустімо, куб) може вільно переміщуватися в просторі і має свободу шостого ступеня, тобто за допомогою

трьох обертальних і трьох поступальних рухів воно може бути переміщене на будь-яку іншу позицію, що для промислового робота відповідає комбінуванню рухів за ротаційними і трансляційними осями. Вихідною для даних процесів є *система координат* із звичайними осями координат x, y, z для осей поступальної ходи за прямими лініями (поступальні рухи за напрямками осей координат). Осі обертальних рухів позначають буквами a, b, c . Можливості виконання окремих рухів передавальним механізмом направляючого грейфера і самого грейфера визначають ступінь свободи робота (який називається також ступенем свободи грейфера). Щодо куба він має свободу шостого порядку. Зображення куба з напрямками поступальних рухів x, y, z та обертальних рухів a, b, c подано на рисунку 1.7.

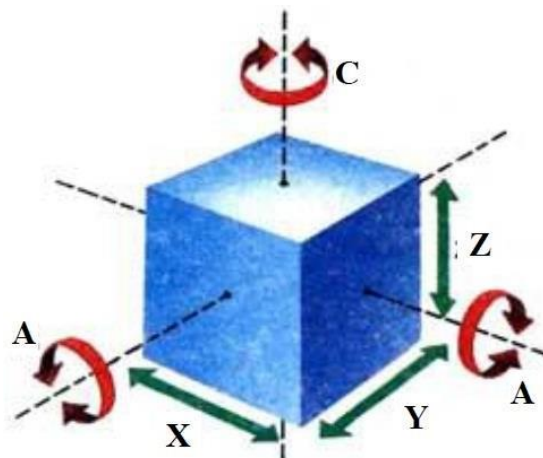


Рисунок 1.7 – Вид куба з напрямками поступальних рухів x, y, z та обертальних рухів a, b, c

Можливість виконання рухів промислового робота забезпечується взаємодією різноманітних шарнірів, з'єднаних між собою робочими ланками.

На рисунку 1.8 подано зображення тримірної моделі робота – маніпулятора.

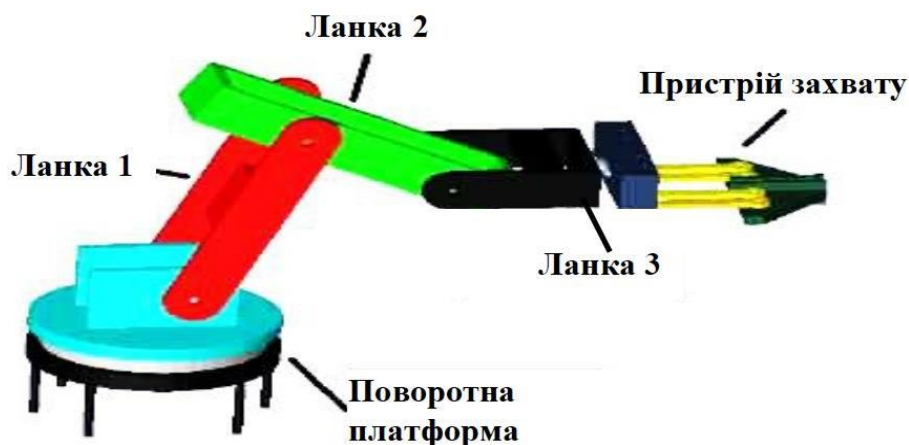


Рисунок 1.8 – Тримірна модель робота–маніпулятора

Так, для переміщення центральної точки грейфера в будь-яку точку робочої зони необхідні три рухи, для надання грейферу в цій позиції бажаного напрямку необхідні ще три рухи. Як правило, три рухи приходяться на важіль і три - на грейфер.

Вибір необхідного ступеня свободи для робота залежить, звичайно, від тих завдань, які належить йому виконувати.

Наприклад, завантажувальний робот, який може обходитися відносно простими рухами, не потребує високого ступеня свободи, на відміну від робота, призначеного для миття і чищення автомобільних кузовів.

1.6 Робочі органи маніпуляторів

Взаємодія робота із зовнішнім середовищем здійснюється, в основному, за допомогою його грейферів, що знаходяться в безпосередньому контакті з обробляючим предметом. Тому конструкція грейфера залежить від виду силової передачі; кількості шарнірів; прохідної відстані, виду поверхні; розмірів і маси об'єкта, а також від виду матеріалу об'єкта і т. п.

Грейфер розміщується на важелі і призначається для втримування інструменту або оброблюючої деталі. За допомогою передавального механізму грейфера, а разом з ним і оброблююча деталь або інструмент переміщуються на позицію, необхідну для виконання певної операції.

Найчастіше грейфери оснащуються додатковими елементами – магнітами, датчиками і т.п.

Робочі органи маніпуляторів діляться на *захватні пристрої* і *спеціальний інструмент*.

Робочі органи можуть бути *постійними* і *знімними*, в тому числі з можливістю їх автоматичної заміни в ході виконання технологічної операції.

Захватні пристрої призначені для того, щоб брати об'єкт, утримувати його в процесі маніпулювання і звільнити після закінчення цього процесу.

Існують такі основні типи захватних пристроїв: *механічні пристрої* – *захвати*, *пневматичні* і *електромагнітні*. Крім того, в зв'язку з великою різноманітністю об'єктів маніпулювання розроблено велику кількість різних комбінацій цих типів захватних пристроїв і безліч спеціальних захватних пристроїв, що базуються на різних оригінальних принципах дії (наприклад, клейкі захватні пристрої, наколюючі, що використовують аеродинамічну підйомну силу і т. п.).

Захвати – це механічні захватні пристрої, які є аналогом кисті руки людини. Найпростіші двопальцеві захвати нагадують звичайні плоскогубці, але забезпечені приводом.

Промислові роботи, в своїй більшості, оснащуються *стандартними* грейферами, у яких можна замінювати окремі елементи.

Для виконання певних робочих операцій іноді доводиться виготовляти *спеціальні* грейфери.

На рисунку 1.9 зображено окремі типи найбільш розповсюджених грейферів у робототехніці.



Рисунок 1.9 – Окремі типи грейферів в робототехніці

Для одночасного виконання різних робочих операцій роботи оснащуються декількома важелями, на кожному з яких встановлюються один або два грейфери. Так, безперерйна робота завантажувальних роботів або завантажувальних пристроїв для турнікетних автоматів забезпечується трьома важелями, відповідно, одним або двома грейферами.

Захвати часто наділяють «чутливістю» за допомогою контактних датчиків (торкання чи тактильних), датчиків прослизання, зусилля (по одній або декількох осях) і дистанційних датчиків (ультразвукових, оптичних та ін.), які виявлятимуть предмети поблизу захвату і між його пальцями.

У найбільш поширеному типі вакуумного загарбного пристрої використані вакуумні присоски, які утримують об'єкти за рахунок розрядження повітря в порожнині між присосками і захоплюваним об'єктом. Для захоплення об'єктів складної форми застосовують вакуумні захватні пристрої з декількома присосками.

Магнітні захватні пристрої також давно відомі, як і вакуумні, і широко використовуються для взяття феромагнітних об'єктів.

У роботах знайшли застосування в основному захватні пристрої з електромагнітами, але є пристрої і з постійними магнітами.

На рисунку 1.10 зображені спеціальні грейфери а) з вакуумним і магнітним захватом, б) з охватом, в) з тактильними датчиками.

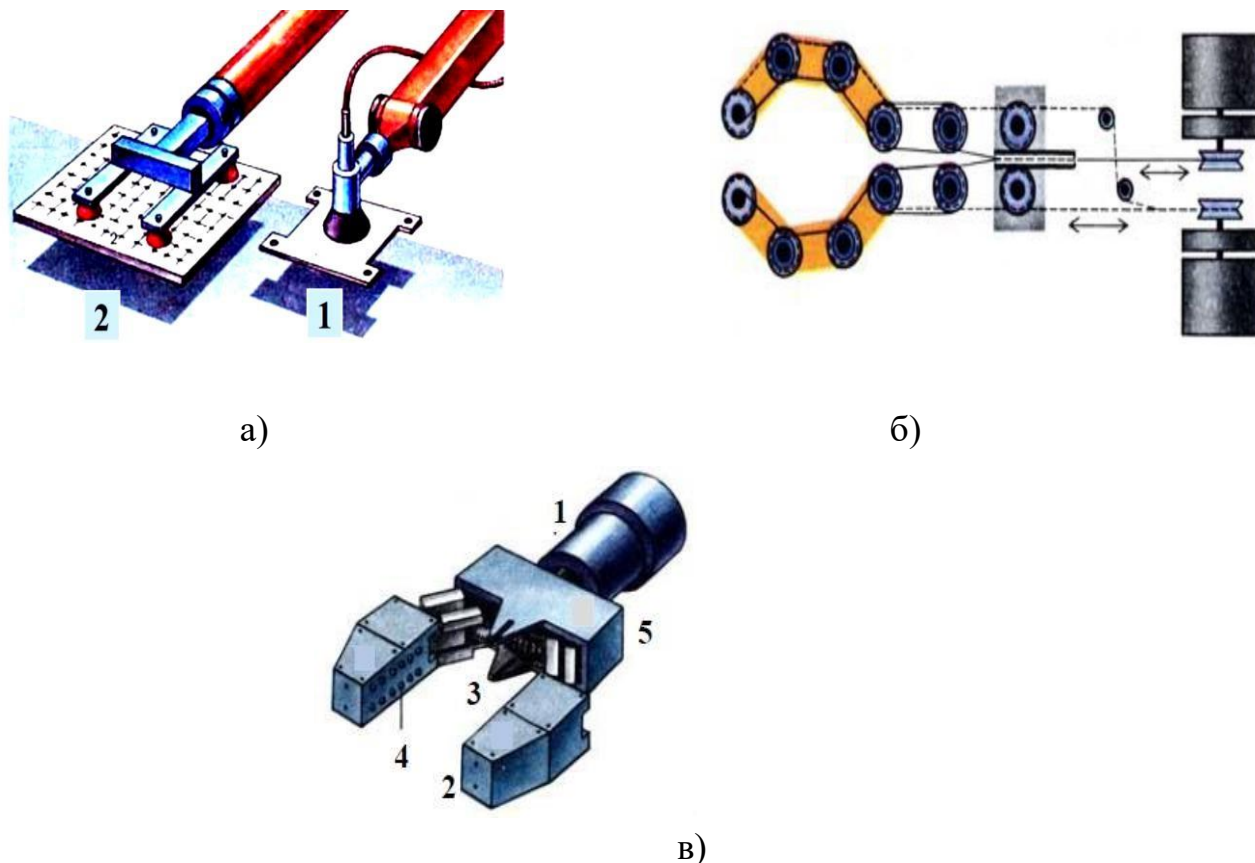


Рисунок 1.10 – Спеціальні грейфери: а) грейфер з вакуумним 1 і магнітним захватом 2; б) грейфер з охватом; в) грейфер з тактильними датчиками; 1 – шарнірний датчик; 2 – датчики доторкання; 3 – тримач інструменту; 4 – грейферні датчики; 5 - кожух приводу

Для звільнення захопленого предмета грейфери зазвичай постачають спеціальними механічними виштовхувачами. Як правило, для приведення в дію грейферів і невеликих маніпуляторів. Іноді для збільшення ефективності приводи комбінуються, що дозволяє домагатися кращих позицій обробляючих деталей і інструменту.

Якщо робот не призначений для операцій захвату об'єкта, то важіль із грейфером може бути замінений на відповідний інструмент для виконання спеціальних технологічних операцій: зварювання, свердління, монтажу, фарбування фарбопультотом і ін.

Часто завданням робота є виконання певних технологічних операцій, таких як: нанесення покриттів, зварювання, загвинчування гайок, зачищення поверхонь і

т.п., коли замість захватного пристрою необхідно використовувати робочий інструмент. При цьому цей інструмент, як правило, безпосередньо кріпиться до маніпулятора.

На рисунку 1.11, для прикладу, зображено робота, який призначений для штампування прокладок.



Рисунок 1.11 – Зовнішній вид робота з інструментальним робочим органом

Ці роботи називають *коботами*. Вони створені спеціально для безпечної роботи біля людини і відзначаються мобільністю, компактністю, зручними засобами ручного керування та іншими перевагами.

ТЕМА 2 СИСТЕМИ КЕРУВАННЯ ПРОМИСЛОВИХ РОБОТІВ

2.1 Класифікація систем керування

Промислові роботи (ПР) можуть класифікуватися відповідно до швидкодії і точності руху. Ці параметри характеризують систему керування ПР, динамічні властивості роботів – взаємозалежні. Швидкодія і точність ПР складається з їх значень для маніпуляторів і пристрою пересування. Головними і специфічними для робототехніки є швидкодія, точність та надійність маніпуляторів.

Швидкодію ПР загального застосування можна розбити на такі групи:

1) низька, за лінійних швидкостей відповідно до окремих ступенів рухомості до 0,5 м/с;

2) середня, за лінійних швидкостей понад 0,5 до 1 м/с;

3) висока, за лінійних швидкостей понад 1 м/с.

Більшість сучасних роботів мають середню швидкодію і тільки 20% їх загальної кількості – високу.

Цей параметр для більшості роботів визначає їх продуктивність. Зазначена швидкодія сучасних роботів є недостатньою і потрібно її збільшити принаймні

вдвічі. Основні труднощі, які виникають при цьому протиріччя швидкодії та точності.

Найчастіше точність ПР характеризується абсолютною похибкою.

Точність ПР загального застосування можна розбити на такі три групи:

- 1) низька, при лінійній похибці від 1 мм і вище;
- 2) середня, від 0,1 до 1 мм;
- 3) висока, менше за 0,1 мм.

Більшість ПР у світі мають середню точність. Однак існують ПР із точністю до одиниць мкн.

Розглянуті далі класифікаційні параметри характеризують технічний рівень ПР.

До них належать і розглянуті раніше параметри, що можуть мати кількісне визначення: швидкодія, точність, об'єм пам'яті, кількість каналів зв'язку із зовнішнім обладнанням. Однак якщо під час використання цих параметрів для класифікації ПР їх розбивають на класифікаційні групи (5 – відповідно до вантажопідйомності, 3 – точності і швидкості і т.п.), то порівняльну оцінку технічного рівня робота виконують виходячи з конкретних числових значень параметрів.

Іншими параметрами, якими характеризується технічний рівень робіт є надійність, кількість одночасно працюючих ступенів рухомості, час програмування, а також похідні від цих параметрів.

Керування руху робіт відповідно до окремих ступенів рухомості може бути контурним і дискретним. Найпростішим варіантом дискретного керування є циклове, за якого кількість точок позиціонування відповідно до кожного ступеня рухомості мінімальна, тобто найчастіше обмежена двома (початковим і кінцевим) координатами.

До важливих параметрів систем керування робіт, що визначають їх експлуатаційні можливості, належать об'єм пам'яті пристроїв керування (ПрКР), типи і кількість каналів зв'язку із зовнішнім устаткуванням і з людиною-оператором, а також призначення і способи взаємодії з оператором.

Об'єм пам'яті ПрКР поряд із загальною оцінкою в Кбітах визначається ще найбільшою кількістю програм, що заносяться у пам'ять.

2.2 Системи програмного керування

Системи програмного керування (СПК) призначені для програмування роботи ПР, запам'ятовування програми керування, її збереження, відтворення, видачі керуючих команд, а також для контролю їх виконання.

Під програмним керуванням розуміється зашифрована тим або іншим способом інформація про послідовність і час виконання окремих керуючих команд робочого циклу, про просторове положення робочих органів ПР.

Системи програмного керування можна класифікувати:

- 1) за способом позиціювання;
- 2) за способом керування: розімкнуті; зімкнені;
- 3) за структурою: з постійною структурою на базі спеціальних обчислень; із змінною структурою на основі мікроконтролера; із прямим керуванням від зовнішньої ЕОМ;
- 4) за способами програмування.

У позиційних системах керування (СКР) задаються початкове і кінцеве положення робочих органів ПР. Розрізняють малоточкові та багатоточкові позиційні СКР.

Малоточкові мають не більш 8-10 точок позиціювання. Багатоточкові – до декількох сотень точок.

У СКР положення робочого органа ПР визначено в кожен момент часу. В електромеханічних контурних СКР геометрична інформація подана у вигляді фізичного аналога (положення упорів, настроювання часу). Інформація про час і послідовність виконання кроків програми (цикл роботи) може задаватися переналагоджуваними схемами релейної автоматики.

У циклових СКР команди циклу задаються у вигляді чисел, а геометрична інформація – упорами і підключенням відповідних вимикачів. Перебудова циклу під час використання внутрішніх комутаторів зводиться до установки перемикачів і т.п. у відповідні гнізда.

Системи програмного керування ПР мають такі відмінності від верстатних.

1. Програмування методом навчання.
2. Значна кількість входів–виходів для зв'язку з основним і допоміжним устаткуванням.
3. Додаткові модулі виміру показників стану механізмів ПР і параметрів зовнішнього середовища.
4. Модулі діагностики для реалізації функцій диспетчеризації і контролю роботи устаткування і пристроїв, приєднані до ПР.
5. Спеціальне математичне забезпечення.
6. Підвищені швидкості та значні переміщення робочих органів.
7. Величини дискретизації.
8. Наявність спеціалізованих циклів (завантаження–розвантаження устаткування).
9. Розпізнавання і вимірювання зовнішніх об'єктів.

10. Адаптивне керування.

Для керування простими низькофункціональними ПР використовуються малоточкові циклові системи позиційного керування типу УЦМ.

Основний принцип циклового позиційного керування ПР полягає в позиціюванні маніпулятора ПР відповідно до упорів або дискретних датчиків.

Це визначає ряд характерних рис ПрКР, головні з яких такі:

1) інформація про переміщення стосовно окремих ступенів рухомості задається за допомогою регульованих або упорів датчиків положення;

2) задане і фактичне положення ланок маніпулятора порівнюються у природному коді;

3) під час програмування технічна і технологічна інформація задаються в дискретному вигляді, вони визначають послідовність руху ланок маніпулятора, тривалість позиціювання і т.д.;

4) керування ведеться за розімкнутим типом.

2.3 Циклові системи керування

Розглянемо циклову систему керування на прикладі системи МПЦП.

МПЦП призначений для циклового двопозиційного керування маніпуляторами та технологічним обладнанням. Він дозволяє здійснювати циклове (за часовим, шляховим або сполучними принципами) та програмно-логічне керування.

МПЦП володіє такими функціями:

1) програмувальні: керування виходами на виконавчий пристрій; прийняття інформації, яка надходить від датчиків стану обладнання; формування витримки часу; керування лічильниками; звернення до підпрограм; організація умовних і безумовних переходів за програмою; зв'язок відповідно до інтерфейсу послідовної передачі інформації;

2) сервісні: редагування програм; тестовий контроль модулів; контроль робочих програм.

Під час введення інформації з клавіатури пульта керування та виводу її на дисплей застосовується шістнадцятирічна система обчислення.

Дискретність завдання витримки часу – 0,1 с. МПЦП забезпечує світлову індикацію стану кожного входу та виходу. Споживана потужність МПЦП – не більше 200 Вт. Маса – не більше 18 кг.

Залежно від об'єму пам'яті робочих програм, кількості входів–виходів, наявності модуля послідовного інтерфейсу (МПП) і типу корпусу випускаються різноманітні МПЦП (табл. 2.1).

Таблиця 2.1 – Види МПЦП, що випускаються

Скорочене позначення	Тип корпусу	Об'єм пам'яті, Кбайт	Кількість входів	Кількість виходів	Наявність МП
МПЦП-1-32-1	Вбудований	1	32	32	Немає
МПЦП-2-32-1	Приладовий	1	32	32	Немає
МПЦП-1-48-2	Вбудований	2	48	48	Є
МПЦП-2-48-2	Приладовий	2	48	48	Є

Параметри сигналів зв'язку МПЦП із зовнішніми пристроями подані в таблиці 2.2.

Таблиця 2.2 – Параметри сигналів зв'язку МПЦП

Параметр	Величина
Напруга вихідного комутуючого сигналу, В	20-30
Максимальний комутуючий струм, А	0,5
Високий рівень вхідного сигналу, В	20-30
Низький рівень вхідного сигналу, В	Не більше 5
Вхідний струм за напруги 24 В, мА	13
Час реакції МПЦП на вхідний сигнал, мс	4
Довжина лінії зв'язку модуля МП, м	До 300

Структурна схема МПЦП наведена на рисунку 2.1.

У склад МПЦП входять:

- 1) система пам'яті, яка включає модуль пам'яті (МПМ) та модуль енергонезалежного запам'ятовуючого пристрою (МЕНЗП) – 512 байт;
- 2) модуль процесора (МПР);
- 3) система вводу–виводу, яка забезпечує за допомогою відповідних модулів зв'язок МПР з ПКР, що керується ЕОМ вищого рангу та зовнішнім технологічним обладнанням;
- 4) система електропостачання.

Модуль процесора здійснює збір, цифрову обробку та виведення інформації відповідно до виконавчої програми, що записана у перепрограмний постійнозапам'ятовуючий пристрій (ППЗП) модуля пам'яті.

Виконавча програма (ВКП) є невід'ємною частиною МПЦП, недоступною для користувача. Її призначення – перетворення команд, які вводяться оператором із ПКР або надходять від керуючої програми (КрП) відповідно до послідовності кодів машинної мови мікропроцесора для реалізації цих команд.

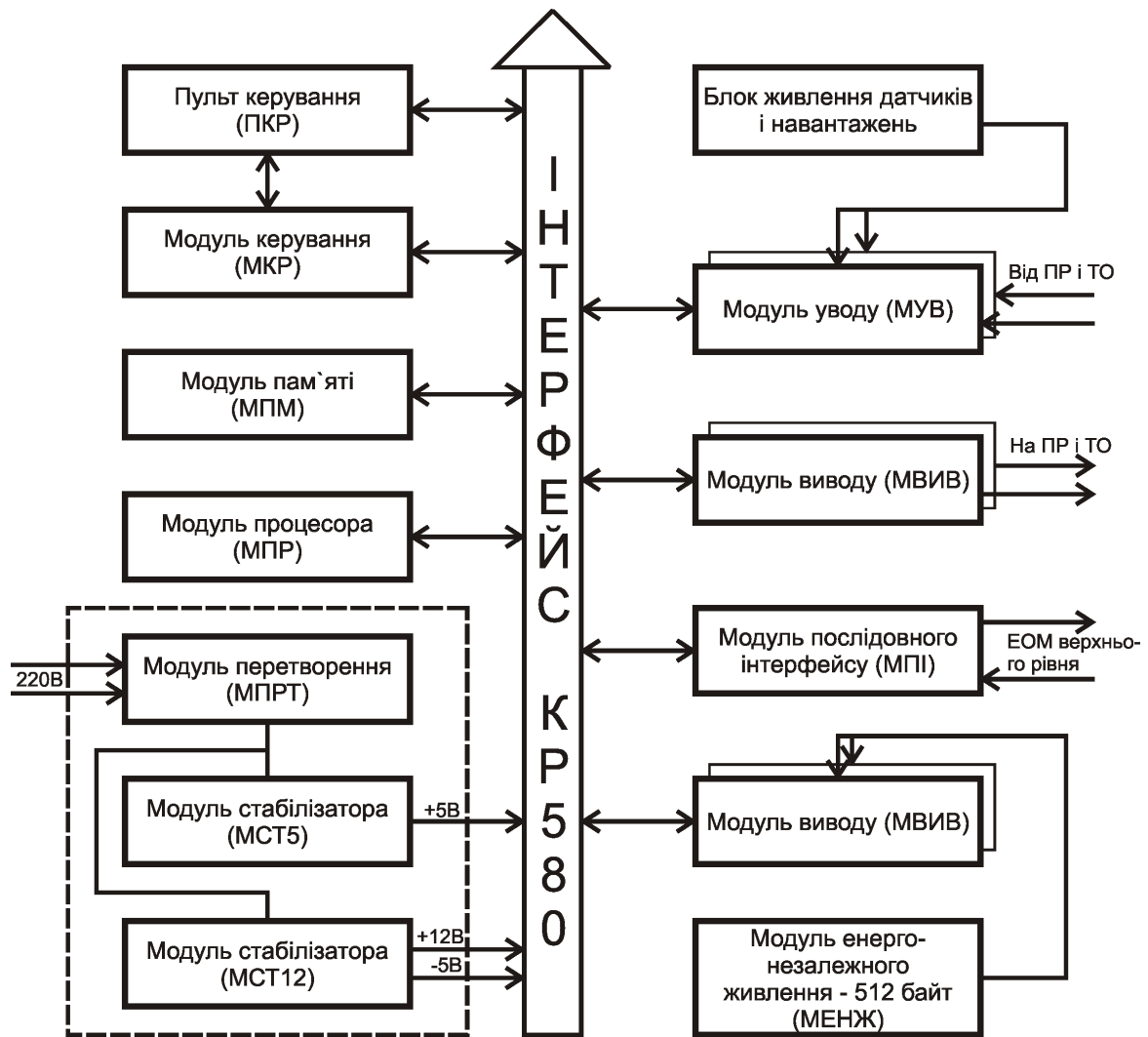


Рисунок 2.1 – Структурна схема МПЦП

Керуюча програма – це програма, написана користувачем у кодах команд вхідної мови МПЦП, що забезпечує поданий алгоритм роботи ПР і технологічного обладнання (ТО). Ця програма розміщується в МЕНЗП та зберігається за відключення напруги завдяки використанню батарей елементів живлення.

Пульт керування спільно з модулем керування (МКР) складають технічні засоби спілкування оператора з МПЦП.

До пульта керування входять: клавіатура для введення команд і керування режимами роботи МПЦП, однорядковий дисплей, призначений для відображення інформації, що контролюється оператором і індикатори режимів роботи, рисунку 2.2.

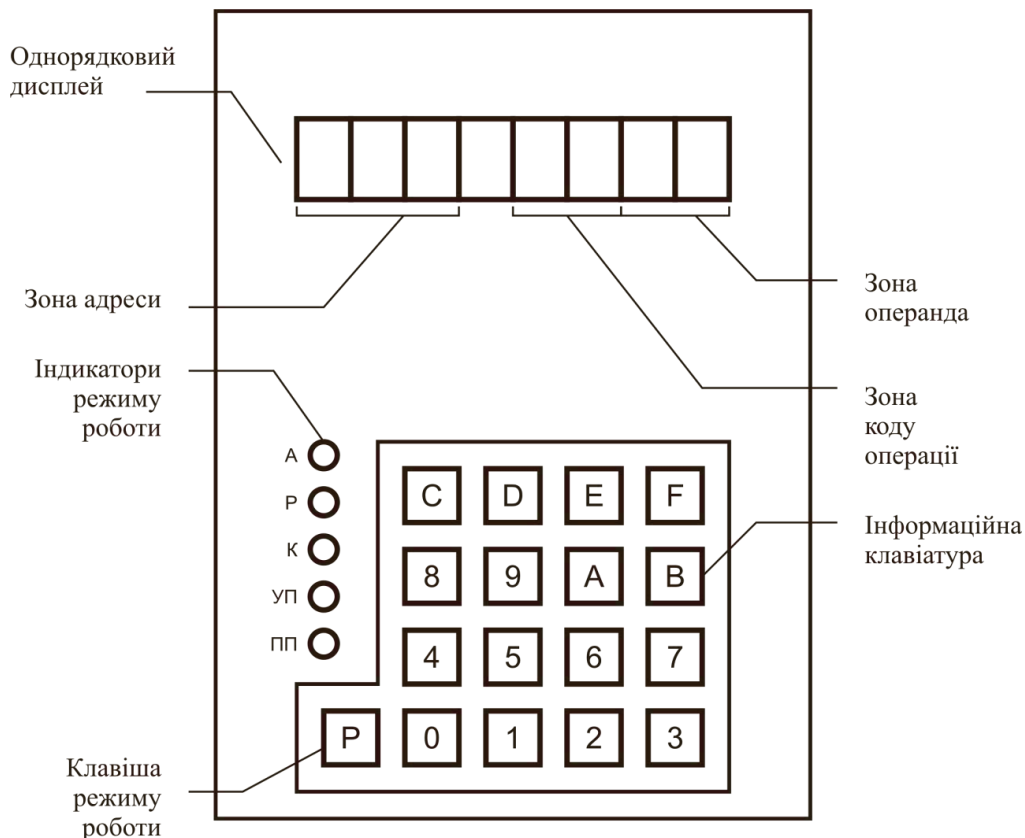


Рисунок 2.2 – Зовнішній вигляд пульта керування

Модуль керування забезпечує з'єднання клавіатури, індикації та пульта керування з внутрішньою магистраллю обміну інформацією МПЦП. Модулі вводу (МУВ) і виводу (МВИВ) дискретних сигналів призначені для зв'язку МПЦП із зовнішнім технологічним обладнанням (реле, датчиками стану обладнання, елементами сигналізації, виконавчими пристроями).

2.4 Режими роботи циклової системи керування

У кожний поточний момент часу МПЦП може знаходитися в одному з таких режимів роботи: автоматичному (А), ручному (Р), кроковому (К), вводу програми (УП), перегляду програми (ПП), функціональної клавіатури.

Узагальнена конфігурація внутрішньої структури МПЦП для цих режимів подана на рисунку 2.3.

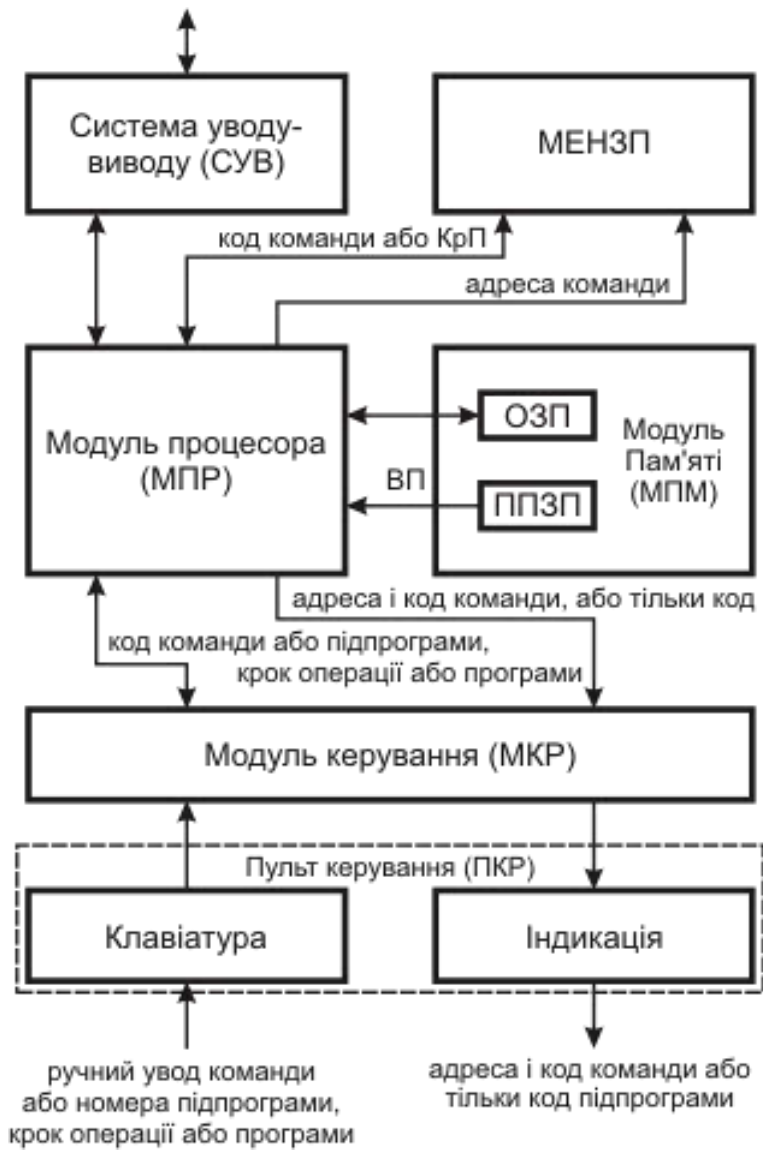


Рисунок 2.3 – Узагальнена конфігурація внутрішньої структури МПЦП

Для зміни режимів роботи необхідно на ПУ натиснути клавішу Р та, не відпускаючи її, натиснути на ще одну клавішу відповідно до таблиці 2.3. При цьому засвітиться один із п'яти індикаторів режиму роботи (рис. 2.2).

Таблиця 2.3 – Приклади перемикання режимів роботи МПЦП

Натиснута клавiша	Індикація режиму роботи					Установлений режим роботи
	А	Р	К	УП	ПП	
«Скидання»	○	●	○	○	○	Ручний
Р0	●	○	○	○	○	Автоматичний
Р1	○	●	○	○	○	Ручний
Р2	○	○	●	○	○	Кроковий
Р3	○	○	○	●	○	Увід програми
Р4	○	○	○	○	●	Перегляд програми вперед
Р5	○	○	○	○	●	Перегляд програми назад
Р6	○	○	●	○	○	Функціональна клавіатура маніпулятора 1
Р7	○	○	●	○	○	Функціональна клавіатура маніпулятора 2
Р8	○	○	○	○	●	Перегляд програми вперед

- – світлодіод увімкнений;
- – світлодіод вимкнений

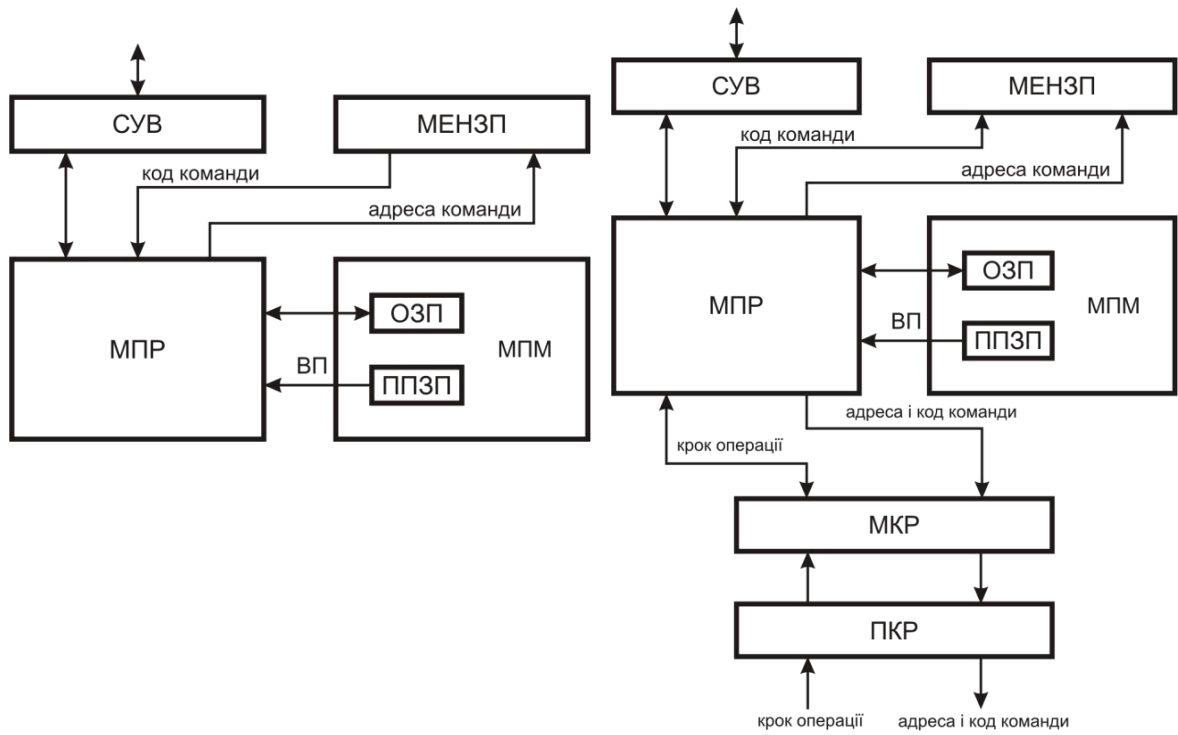
Режим роботи «Автоматичний» – основний і призначений для керування ПР та ТО відповідно до КрП, уведеної у МЕНЗП (рис. 2.4, а).

Для підвищення ефективності відлагодження КрП використовується режим роботи «Кроковий», у якому процесор здійснює вибірку із МЕНЗП та виконання чергової команди (рис. 2.4, б).

У цьому режимі оператор має можливість здійснити виконання КрП у необхідному йому темпі. У паузах між виконанням команд процесор виводить на індикацію адресу (вміст лічильника команд) і вміст чарунки МЕНЗП, яка зберігає команду, що буде виконуватися на наступному кроці.

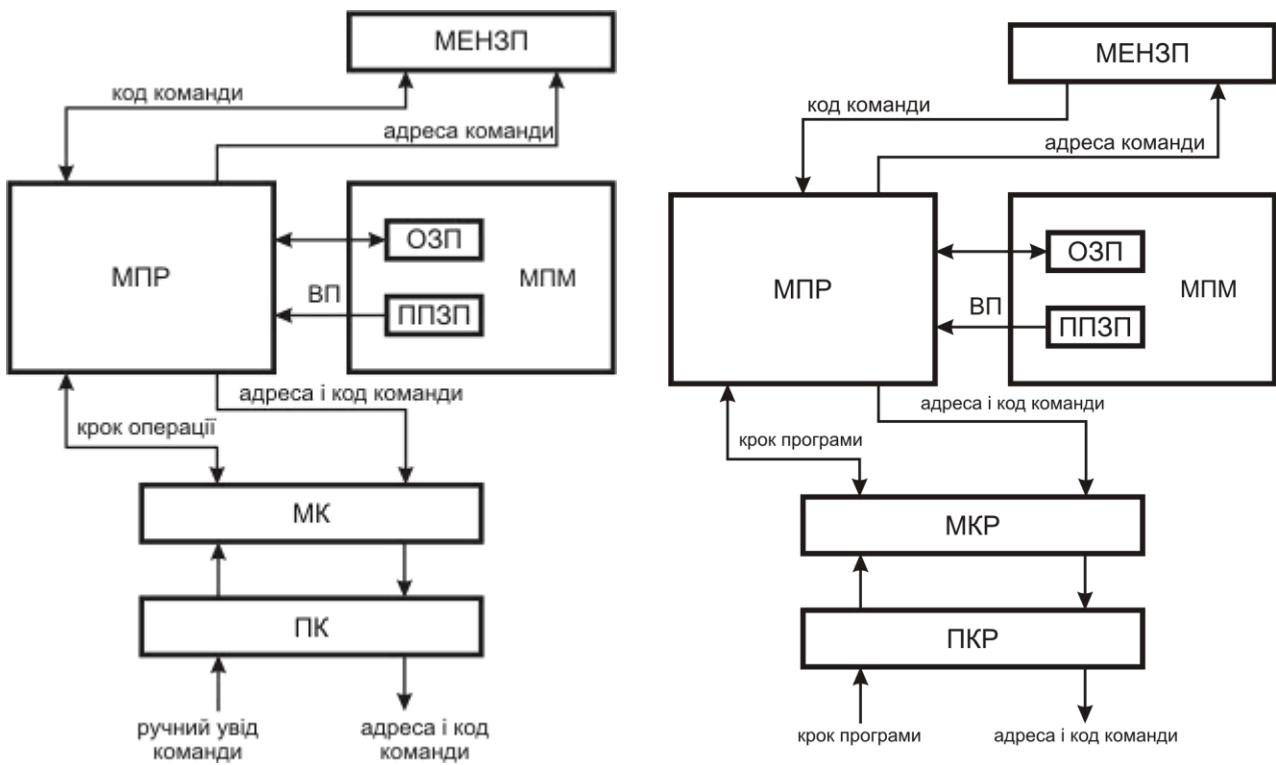
Режим «Увод програми» використовується для запису кодів команд керуючої програми в МЕНЗП (рис. 2.4, в). Необхідна команда набирається оператором на клавіатурі пульта керування.

При цьому процесор під дією ВКП зчитує коди натиснутих клавiш, формує з них код команди і пересилає його в МЕНЗП за адресою, визначеною лічильником команд.



а)

б)



в)

г)

Рисунок 2.4 - Конфігурації внутрішньої структури МПЦП для різних режимів роботи: а – автоматичний; б – кроковий; в – увід програми; г – перегляд програми

Після закінчення пересилки кожної команди вміст лічильника команд автоматично збільшується на одиницю. Інформація, яка вводиться спільно з поточним значенням лічильника команд, відображається на дисплеї пульта керування.

Режим «Перегляд програми» аналогічний режиму «Увод програми», проте напрямок проходження інформації – протилежний (рис. 2.4, г).

Відповідно до адреси чарунки МЕНЗП, записаної у лічильнику команд, процесор зчитує інформацію з чарунки та виводить її на індикацію спільно з поточним значенням лічильника команд.

Таким чином, МЕНЗП є джерелом команд для процесора, вибірку яких він здійснює за адресою, що визначається лічильником команд. Цей режим використовується для контролю оператора над керуючою програмою, яка зберігається у пам'яті мікроконтролера.

У режимі «Ручний» процесор взаємодіє з пультом керування аналогічно тому, як у режимі «Увод програми» (рис. 2.5, а). Сформований код команди у цьому випадку не запам'ятовується в МЕНЗП, а виконується.

Режим керування «Функціональна клавіатура» є модифікацією режиму «Ручний» і спрощує керування маніпулятором під час налагодження РТК (рис. 2.5, б).

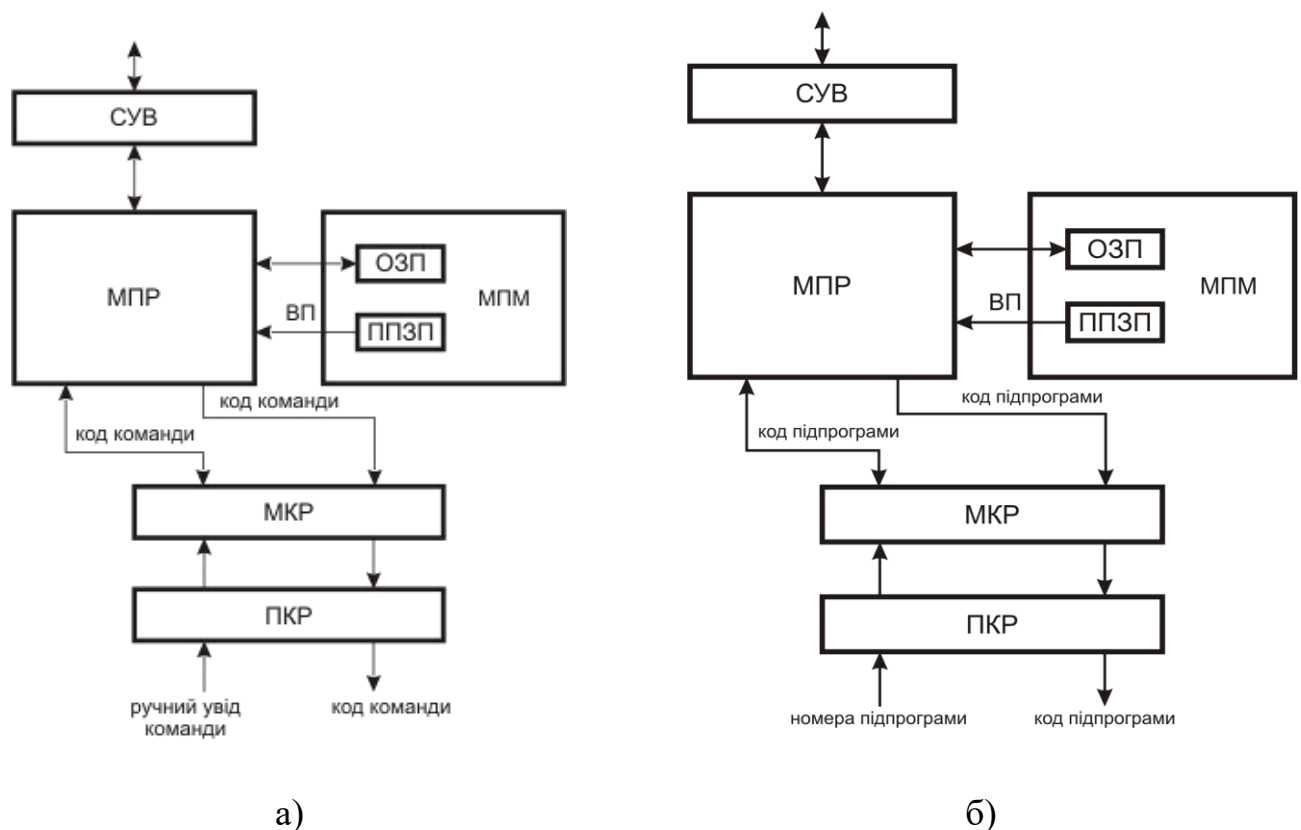


Рисунок 2.5 - Конфігурації внутрішньої структури МПЦП для режимів роботи:
а) «Ручний», б) «Функціональна клавіатура»

Режим «Функціональна клавіатура» дозволяє керувати маніпулятором, електромагніти пневморозподільників якого підключені до виходів 00–09.

Контроль за виконанням руху здійснюється датчиками, приєднаними до виходів 00–09.

Для роботи достатньо перейти у режим «Функціональна клавіатура» (табл. 2.3) і натискати клавіші відповідно таблиці 2.4 (команди режиму «Функціональна клавіатура» 6, 7, 8, 9 у поданому маніпуляторі МП-9С не використовується у зв'язку з відсутністю 4-го та 5-го ступенів рухомості).

У цьому режимі інформаційні клавіші 0-В викликають з ППЗП підпрограми керування рухом маніпулятора.

Під час переходу в режим «Функціональна клавіатура» на однорядковому дисплеї ПУ з'явиться зображення «— — — — F — 0 0», де «—» позначає, що елемент погашений.

Під час натискання кожної клавіші відповідно до таблиці 2.4 на місці 2-го елемента коду операції з'явиться її значення.

Наприклад, при натисненні клавіші 2:

«— — — — F 2 0 0»

Таблиця 2.4 – Підключення датчиків та електромагнітів

Натиснута клавіша	Умовне найменування руху	Дія	Датчик (№ входу)	Електромагніт (№ виходу)
0	1-1	Переведення ступеня рухомості 1 у положення 1	x1(00)	y1(00)
1	1-2	Переведення ступеня рухомості 1 у положення 2	x2(01)	y2(01)
2	2-1	Переведення ступеня рухомості 2 у положення 1	x3(02)	y3(02)
3	2-2	Переведення ступеня рухомості 2 у положення 2	x4(03)	y4(03)
4	3-1	Переведення ступеня рухомості 3 у положення 1	x5(04)	y5(04)
5	3-2	Переведення ступеня рухомості 3 у положення 2	x6(05)	y6(05)
A	6-1	Розтиснення захоплювача	—	y7(06)
B	6-2	Стискування захоплювача	—	y7(06)

Після закінчення виконання відповідного руху цей елемент гасне, тобто з'являється початкове зображення.

ТЕМА 3 МЕХАТРОНІКА І МЕХАТРОННІ СИСТЕМИ

3.1 Класифікація мехатронних об'єктів

Узагальнюючим поняттям мехатроніки є *мехатронний об'єкт* (мо). За визначенням мехатроніки мо являє собою інтеграцію механічних, електротехнічних, мікроелектронних і комп'ютерних компонентів.

Мехатронні об'єкти залежно від рівня інтеграції компонентів і ступеня складності та з урахуванням термінів, що виникли у сфері промислової робототехніки і гнучких виробничих систем, можна класифікувати як: мехатронні модулі (вузли), агрегати (машини), комплекси і системи.

Мехатронний модуль (мм) – це мехатронний об'єкт, призначений, як правило, для реалізації рухів за однією координатою.

Мехатронний агрегат (ма) включає декілька мехатронних модулів і призначений для реалізації заданих законів руху за декількома координатами в умовах взаємодії із зовнішнім середовищем.

Мехатронний комплекс складається з декількох агрегатів або агрегату і ряду окремих модулів, зв'язаних між собою і діючих як одне ціле.

Мехатронна система (мс) – це об'єднання декількох комплексів або комплексу та інших мо на базі єдиної інтеграційної платформи.

Якщо розглядати появу мо на ринку наукоємної продукції історично, то їх можна розділяти за поколіннями.

Мо першого покоління об'єднували тільки два початкових компоненти і були практично мм (хоча ще не повністю відповідали за визначенням мо). Типовим прикладом мо першого покоління може бути "мотор-редуктор", де механічний редуктор і керований двигун випускаються як єдиний функціональний елемент, тобто мм. Мехатронні агрегати на основі цих модулів знайшли широке застосування під час створення різних засобів комплексної автоматизації виробництва (конвеєрів, транспортерів, поворотних столів, допоміжних маніпуляторів).

Мо другого покоління з'явилися у 80-х роках ХХ ст. У зв'язку з розвитком нових електронних технологій, які дозволили створити мініатюрні датчики й електронні блоки для обробки їх сигналів. Об'єднання привідних модулів із вказаними елементами привело до появи мехатронних модулів руху, склад яких повністю відповідає назві, коли досягнута інтеграція трьох пристроїв різної фізичної природи: механічних, електротехнічних і електронних. На базі

мехатронних модулів даного класу створені керовані енергетичні агрегати (турбіни й генератори), верстати і промислові роботи з числовим програмним керуванням.

Розвиток третього покоління мо обумовлений появою на ринку порівняно недорогих мікропроцесорів і контролерів на їх базі та направлений на інтелектуалізацію всіх процесів, що протікають в мехатронній системі, у першу чергу – процесу керування функціональними рухами агрегатів і комплексів. Одночасно йде розробка нових принципів і технологій виготовлення високоточних і компактних механічних вузлів, а також нових типів електродвигунів (у першу чергу високомоментних, безколекторних і лінійних), датчиків і інформаційних технологій. Синтез нових прецизійних, інформаційних і вимірювальних наукоємних технологій дає основу для проектування та виготовлення інтелектуальних мехатронних агрегатів і комплексів.

Надалі у четвертому поколінні мехатронні агрегати і комплекси об'єднуюватимуться в мехатронні системи на базі єдиних інтеграційних платформ. Мета створення таких систем – добитися поєднання високої продуктивності та одночасно гнучкості техніко-технологічного середовища за рахунок можливості їх реконфігурації. Це дозволить забезпечити конкурентоспроможність і високу якість продукції, що випускається, на ринках ххі століття.

Інтелектуальні системи керування мо мають такі шари обробки невизначеної інформації.

1. Інтерактивний людино-машинний діалог.
2. Прогноз подій.
3. Самонавчання й адаптація.
4. Робота з базами подій, знань і формування рішень.
5. Старанний.

Ступінь інтелектуальності системи керування мо залежить від того, скільки шарів має та або інша система. Відповідно до ступеня інтелектуальності системи керування мо поділяються на:

- 1) інтелектуальні в малому,
- 2) інтелектуальні у великому,
- 3) інтелектуальні в цілому.

У першому випадку функціонування обмежене двома, у другому – трьома, а у третьому – всіма шарами інтелектуальності. При цьому під інтелектуальними системами керування мо розуміється клас систем, що будуються із застосуванням нових інформаційних технологій обробки і використання знань.

Такий підхід до синтезу систем керування дозволяє підвищити динамічні характеристики створеної системи. Це реалізується шляхом лінгвістичної

апроксимації поведінкових характеристик керованого об'єкта і відмовою від традиційного зворотного зв'язку в системі керування (за умови можливості адекватно подати роботу системи за допомогою знань на основі певних правил).

Проте, якщо апроксимація не вдалася, то інтелектуальний регулятор може мати гірші динамічні характеристики порівняно із звичайними (наприклад з підрегулятором).

Мо можуть мати постійну структуру, тобто бути детермінованими чи бути недетермінованими, отже, такими, що адаптуються до зовнішніх умов. При цьому можуть змінюватися не лише параметри, але і структура об'єкта. Для цього у структурі передбачаються ланки, які за певних умов включаються в роботу. Недетерміновані мо – це інтелектуально-керовані об'єкти, оскільки прийняття рішення щодо зміни структури проводиться лише під час аналізу ситуації.

Відповідно до типу зв'язку з еом верхнього рівня чи інформаційної підтримки мо класифікують:

- 1) з провідним зв'язком;
- 2) з безпроводним зв'язком (радіоканал, інфрачервоний канал, ультразвуковий канал).

Мо (аналогічно з пр) можна класифікувати так:

- 1) відповідно до характеру виконуваних операцій: технологічні (виробничі); допоміжні (підйоми-транспортні); універсальні;
- 2) відповідно до сфери застосування: автомобілебудування, верстатобудування; залізничний транспорт; устаткування для автоматизації технологічних процесів; медичне устаткування; устаткування шоу-індустрії; авіаційна, космічна і військова техніка; робототехніка; обчислювальна й офісна техніка, фото- та відеотехніка; побутова техніка; тренажери для підготовки операторів складних технічних систем, водіїв і пілотів; нетрадиційні транспортні засоби; контрольно-вимірні пристрої і машини; поліграфічні машини;
- 3) відповідно до мобільності: стаціонарні (вбудовані в обладнання, підлогові та підвісні); пересувні (наземні, підлогові й підвісні).

3.2 Функції мехатроніки

Мехатроніка використовується у багатьох сучасних пристроях, агрегатах і машинах. Наприклад, ПК містить масу мехатронних агрегатів: накопичувачі на гнучких дисках, жорсткі диски, cd-drives, сучасні накопичувачі на магнітних стрічках.

Мехатронні системи також присутні у сучасних пральних машинах-автоматах, пилососах, автомобілях і т.п. Типова мехатронна система – гальмівна

система автомобіля з абс (антиблокувальною системою), що складається з десятків мм і мехатронних агрегатів.

У промисловості мехатронними системами є всі сучасні роботи, верстати, вимірювальні комплекси. Такі системи складаються з декількох приводів, кожен із яких, у свою чергу є мехатронним модулем.

Об'єми світового виробництва мо щорічно збільшуються, охоплюючи все нові сфери. Сьогодні мехатронні модулі і системи знаходять широке застосування у таких пристроях як:

1. антиблокувальних системах гальм, системах стабілізації руху автомобіля й автоматичної парковки;
2. системах контролю і стабілізації руху потягів;
3. реабілітаційному, клінічному та сервісному медичному устаткуванні (апарати типу "штучне серце", "штучна нирка", керовані біоелектричні й екзоскелетні протези для інвалідів);
4. електровелосипедах, вантажних візках, інвалідних колясках;
5. принтерах, плотерах, дисководах;
6. копіювальних і факсимільних апаратах;
7. пральних, посудомийних, швейних і інших побутових машинах.

Стрімкий розвиток мехатроніки, як нового науково-технічного напрямку, обумовлений такими основними чинниками як:

1. розвитком фундаментальних основ і методології мехатроніки (базові наукові ідеї, принципово нові технічні та технологічні рішення);
2. новими тенденціями світового індустріального розвитку;
3. активністю фахівців у науково-дослідній і освітній сферах.

Можна виділити такі тенденції зміни й ключові вимоги світового ринку в даній галузі:

1. інтернаціоналізація ринку науково-технічної продукції і, як наслідок, необхідність активного впровадження у практику форм і методів міжнародного інжинірингу і трансферу технологій;
2. бурхливий розвиток комп'ютерних систем і технологій, засобів телекомунікації (прямим слідством цієї загальної тенденції є інтелектуалізація систем керування механічним рухом і технологічними функціями сучасних машин);
3. необхідність випуску та обслуговування устаткування відповідно до міжнародної системи стандартів якості;
4. підвищення ролі малих і середніх виробничих підприємств у економіці завдяки їх здібності до швидкого та гнучкого реагування на вимоги ринку.

Аналіз наведених тенденцій показує, що досягти якісно нового рівня основного технологічного устаткування на основі традиційних підходів вже практично нереально.

Розвиток мехатроніки ставить нові організаційно-економічні проблеми. Сучасні підприємства, що розпочинають розробку і випуск мехатронних виробів, повинні вирішити в цьому плані чимало основних завдань, а саме:

1. структурну інтеграцію підрозділів механічного, електронного та інформаційного профілів (які, як правило, функціонували автономно) в єдині проектні й виробничі колективи;

2. підготовку «мехатронно-орієнтованих» інженерів і менеджерів, здатних до системної інтеграції та керування роботою вузько профільних фахівців різної кваліфікації;

3. інтеграцію інформаційних технологій з різних науково-технічних галузей (механіка, електроніка, комп'ютерне керування) в єдиний інструментарій для комп'ютерної підтримки мехатронних завдань;

4. стандартизацію та уніфікацію всіх використовуваних елементів і процесів під час проектування та виробництва мехатронних систем.

3.3 Концепція проектування мехатронних систем

Розглянемо узагальнену структуру машини з комп'ютерним керуванням (рис. 3.3). В основу побудови даної схеми покладена структура промислових роботів.

Зовнішнім середовищем для машин даного класу є технологічне середовище, яке містить різне основне і допоміжне устаткування, технологічне оснащення та об'єкти робіт. Під час виконання мехатронною системою заданого функціонального руху об'єкти робіт надають збурюючої дії робочому органу. Прикладами таких дій можуть слугувати сили різання для операцій механообробки, контактні сили і моменти сил під час збирання, сила реакції струменя рідини під час операції гідравлічного різання.

Зовнішні середовища можна розділити на два основні класи: детерміновані та недетерміновані. Детерміновані – це середовища, для яких параметри збурюючих дій і характеристики робіт можуть бути наперед визначені з необхідним для проектування мехатронної системи ступенем точності. Деякі середовища є недетермінованими за своєю природою. Це середовища з екстремальними характеристиками: космічні, підводні, підземні і т.п.

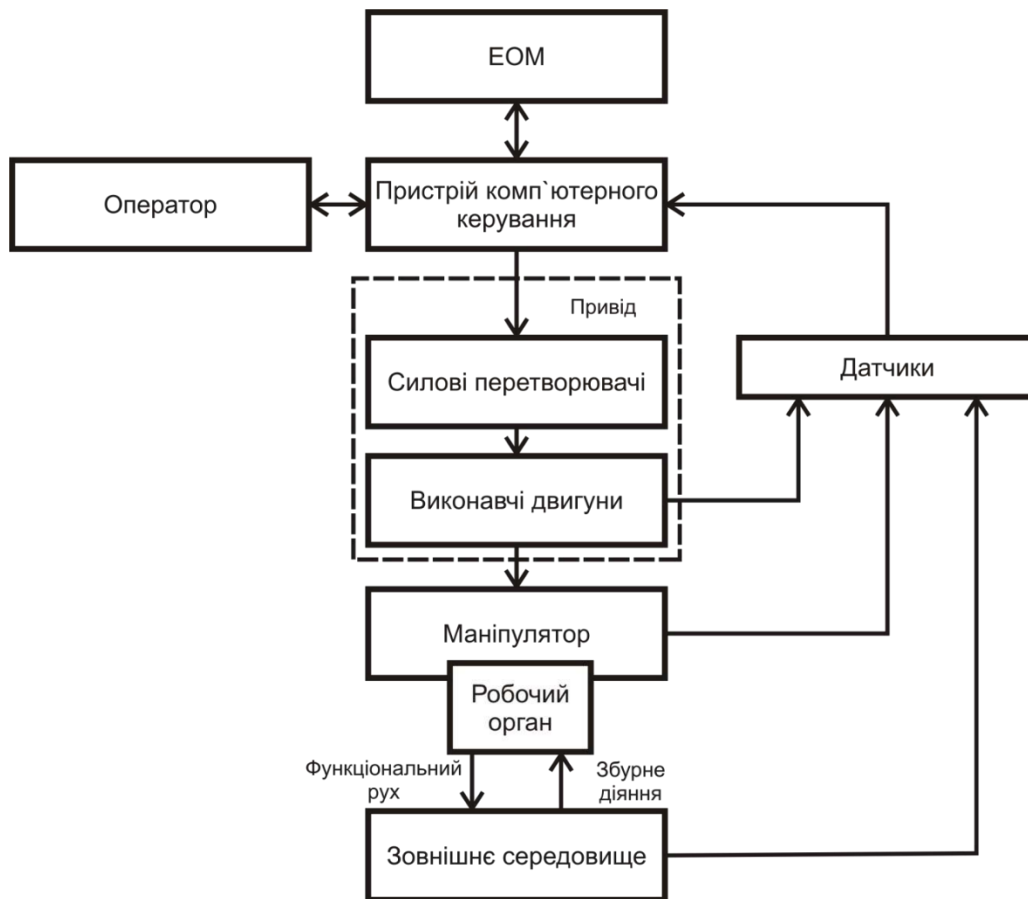


Рисунок 3.3 – Узагальнена структура машини з комп'ютерним керуванням

Характеристики технологічних середовищ, як правило, можуть бути визначені за допомогою аналітико-експериментальних досліджень і методів комп'ютерного моделювання. Наприклад, для оцінки сил різання під час механообробки проводять серії експериментів на спеціальних дослідницьких установках, параметри вібраційних дій вимірюють на вібростендах із подальшим формуванням математичних і імітаційних моделей збурюючих дій на основі експериментальних даних.

Проте для організації і проведення подібних досліджень потрібні дуже складна і дорога апаратура та вимірювальні технології. Так, для попередньої оцінки силових дій на робочий орган під час операції роботизованого видалення облою з литих заготовок необхідно вимірювати фактичні форму і розміри кожної. У цих випадках доцільно застосовувати методи адаптивного керування, які дозволяють автоматично коректувати закон руху мехатронною системою безпосередньо в ході виконання операції.

До складу традиційної машини входять такі основні компоненти:

1. механічний пристрій (маніпулятор), кінцевою ланкою якого є робочий орган;

2. привід, що включає силові перетворювачі та виконавчі двигуни;
3. пристрій комп'ютерного керування, верхнім рівнем для якого є людина-оператор, або інша ЕОМ, підключена через локальну мережу;
4. датчики, призначені для передачі у пристрій керування інформації про фактичний стан блоків машини і стан зовнішнього середовища.

Таким чином, є наявність трьох обов'язкових компонентів: механічного (електромеханічного), електронного і комп'ютерного. Вони зв'язані енергетичними та інформаційними потоками, що також є необхідною ознакою мехатронної системи.

Електромеханічна частина включає механічні ланки і передачі, робочий орган, електродвигуни, датчики й додаткові електротехнічні елементи (гальма, муфти).

Механічний пристрій призначений для перетворення рухів ланок у необхідний рух робочого органу. Електронна частина складається з мікроелектронних пристроїв, силових перетворювачів і електроніки вимірювальних ланцюгів. Датчики призначені для збору даних про фактичний стан зовнішнього середовища та об'єктів робіт, механічного пристрою і блоку приводів із подальшою первинною обробкою й передачею цієї інформації у пристрій комп'ютерного керування. До складу пристрою комп'ютерного керування мехатронної системи звичайно входять комп'ютер верхнього рівня і мікроконтролери керування рухом.

Пристрій комп'ютерного керування виконує такі основні функції:

1. керування процесом механічного руху ММ або МС у реальному часі з обробкою інформації від датчиків;
2. організація керування функціональними рухами, яка припускає координацію керування механічним рухом мехатронної системи і 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 і безпосередньо в процесі руху мехатронної системи;
4. організація обміну даними з периферійними пристроями, датчиками та іншими пристроями системи.

Завданням мехатронної системи є перетворення вхідної інформації, що надходить із верхнього рівня керування, в цілеспрямований механічний рух. Характерно, що електрична енергія (рідше гідравлічна чи пневматична) використовується в сучасних системах як проміжна енергетична форма.

Суть мехатронного підходу до проектування полягає в інтеграції в єдиний функціональний модуль двох або більше компонентів незалежно від їх фізичної природи.

Крім цього, інтегровані мехатронні компоненти вибираються розробником і об'єднуються в єдиний МО вже на стадії проектування машини. У цьому основна відмінність МО від традиційних, у яких проектувальник об'єднує в систему різноманітні механічні, електронні та інформаційні пристрої різних виробників, спроектовані роздільно. Саме тому багато комплексів (наприклад, деякі гнучкі виробничі системи в машинобудуванні) показали на практиці низьку надійність і техніко-економічну ефективність.

Методологічною основою розробки мехатронних систем слугують методи паралельного проектування. За традиційного проектування машин із комп'ютерним керуванням проводиться розробка механічної, електронної і комп'ютерної частин системи, а потім вибір інтерфейсних блоків. Тому на стадії проектування МО з традиційної структури машини виключається принаймні один інтерфейс під час збереження фізичної суті даного модуля. Мета паралельного проектування полягає в одночасному і взаємозв'язаному синтезі всіх компонентів системи.

В ідеальному для користувача варіанті ММ, одержавши на вхід інформацію про мету керування, виконуватиме з бажаними показниками якості заданий функціональний рух. Апаратне об'єднання елементів у єдині конструктивні модулі повинне обов'язково супроводжуватися розробкою інтегрованого програмного забезпечення. Програмні засоби МС повинні забезпечувати безпосередній перехід від задуму системи через її математичне моделювання до керування функціональним рухом у реальному часі.

Базовими об'єктами проектування мехатроніки є ММ, які виконують рухи, як правило, за однією керованою координатою. З таких модулів конструюються мехатронні системи модульної архітектури.

Застосування мехатронного підходу під час створення машин із комп'ютерним керуванням визначає їх основні переваги порівняно з традиційними засобами автоматизації:

1. висока якість реалізації складних і точних рухів внаслідок застосування методів інтелектуального керування;
2. висока надійність, довговічність і перешкодозахищеність;
3. відносно низька вартість завдяки високому ступеню інтеграції, уніфікації і стандартизації всіх елементів та інтерфейсів;
4. конструктивна компактність модулів (аж до мініатюризації в мікромашинах);

5. покращені вагогабаритні й динамічні характеристики машин внаслідок спрощення чи усунення кінематичних ланцюгів;

6. можливість комплексування функціональних модулів у складні системи і комплекси під конкретні завдання замовника.

3.4 Сучасні методи керування мехатронними системами

Однією з цілей мехатроніки є реалізація заданого керованого руху, тому проблеми керування вважаються одними з ключових проблем мехатроніки. Створення якісно нового покоління об'єктів мехатроніки вимагає вирішення широкого спектра складних завдань (забезпечення динамічної розв'язки швидкодіючих приводів із урахуванням компенсації взаємного впливу окремих ступенів рухомості, створення швидкодіючих алгоритмів керування, інваріантного до типу кінематичної схеми багатоланкового МО, вирішення некоректних завдань, облік чинників невизначеності і т.д.).

Класичні принципи теорії керування не забезпечують вирішення цих завдань з таких причин:

1. чимало джерел інформації про різномірні фізичні величини (при цьому датчики розподілені у просторі);

2. великий об'єм обчислень, що не дозволяє багатоканальній системі працювати у режимі реального часу;

3. брак інформації для ухвалення аналітичного рішення;

4. вхідна інформація містить перешкоди і невизначеності.

У зв'язку з цим на перший план виходять нові технології керування. Вони засновані на обробці постійно змінних знань про об'єкт і *називаються інтелектуальними*. Вони є базою самонавчання, що важливо як для автономних об'єктів, так і для об'єктів, що функціонують в умовах безлюдних технологій.

До розряду інтелектуальних технологій можуть бути віднесені такі інформаційні технології, як: експертні системи, системи нечіткої логіки, нейромережні структури, системи з асоціативною пам'яттю. Їх можна доповнити еволюційними (генетичними) алгоритмами самоорганізації.

У даний час серед робіт із створення високоякісних електроприводів найбільший розвиток одержали технології експертних систем і нейромережних структур. У такій системі експертна оцінка виконує функції інтелектуальної надбудови над відомим ПД-регулятором і періодично підстроює його коефіцієнти залежно від зміни параметрів слідкуючого приводу. Експертний регулятор, займаючи об'єм пам'яті близько 350 Кб, забезпечує адаптивне керування в широкому діапазоні збурень, але не володіє швидкодією, необхідною для керування в реальному масштабі часу. Нейромережний регулятор, побудований на

базі 80 статичних нейронів і настроєний на оптимальний відповідно до швидкодії принцип функціонування, включається у контур системи керування послідовно з об'єктом. Він забезпечує дуже високу швидкість під час стеження за різними вхідними діями і, що особливо цікаво, інваріантність до певного типу зовнішніх збурень. Перспективним є застосування технології асоціативної пам'яті, оскільки реалізований на ній інтелектуальний регулятор приводу вимагає менше 20 Кб пам'яті.

Відомі приклади побудови гібридних регуляторів, наприклад, на базі як нечіткої логіки, так і асоціативної пам'яті. Причому застосування останньої технології дозволяє використовувати реальний масштаб часу. Інший гібридний регулятор – нейромережний, початкова модель його модернізована введенням механізму, що попереджає перемикання для компенсації запізнення під час реалізації регулятора на ПВМ і областях лінійного керування в околі нуля фазових координат помилки. Ця модель використовує технологію асоціативної пам'яті і також призначена для застосування у реальному часі.

Усе викладене має велику актуальність для вирішення завдань проектування систем інтелектуального керування з урахуванням суперечності вимог, що постають перед ними.

Зараз застосування методів і технологій обробки знань у завданнях керування на стику штучного інтелекту і теорії керування формуються в новий самостійний науковий напрям інтелектуального керування. Саме на базі інтелектуального керування з'являється можливість створення принципово нового покоління техніки – інтелектуальних мехатронних систем, призначених не тільки для автономного функціонування в умовах невизначеної інформації та організації нових технологічних процесів і виробництв, але й для створення зразків техніки, можливості якої важко передбачити.

Використання перерахованих компонентів штучного інтелекту в мехатроніці дозволяє розширити застосування традиційних методів теорії керування. Механічні рухи МО, як правило, істотно нелінійні, а досягнення необхідної точності здійснюється використанням високоефективних мікропроцесорних систем керування, зокрема адаптивних.

Надзвичайно важливий і найбільш поширений у даний час підклас інтелектуальних систем – нечіткі системи керування (без самонавчання, прогнозування та інтерактивного діалогу).

Використання методів побудови адаптивних структур керування механічними багатоступеневими об'єктами дозволяє не вимагати точного знання всіх подробиць нелінійного опису об'єкта керування, а самі структури при цьому приймають більш універсальний характер і відрізняються меншим об'ємом

обчислень і, як наслідок, простотою реалізації на базі сучасних обчислювальних засобів.

ТЕМА 4 МОДУЛІ РУХУ В МЕХАТРОННИХ СИСТЕМАХ РОБОТІВ

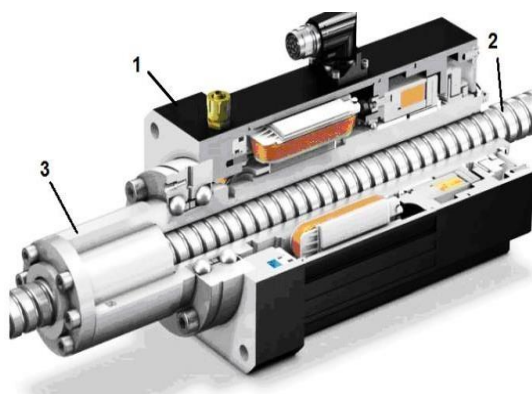
4.1 Поняття модуля руху

Створення мехатронних модулів руху з високими характеристиками є однією з найважливіших задач роботів-маніпуляторів. З таких модулів, як із функціональних блоків, можна створити складні роботизовані системи.

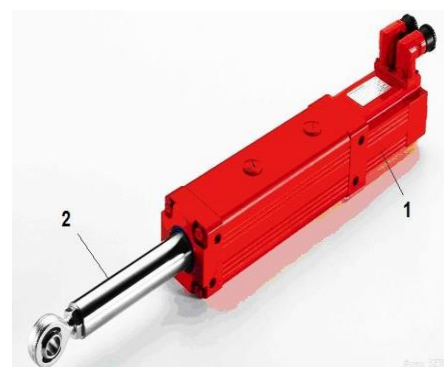
Основними вимогами, щодо будови мехатронних модулів руху для багатьох областей їх застосування є: підвищення продуктивності технологічного обладнання, надійність і компактність.

Модуль руху (MP) – конструктивно й функціонально самостійний виріб, що включає в себе механічну (гідравлічну, пневматичну) і електротехнічну частини, який можна використовувати у різних комбінаціях з іншими модулями і самостійно. Прикладами MP є мотор-редуктори, електроциліндри, мотор-колеса, електрошпинделі. Одним із найперших таких модулів став свого часу мотор-редуктор, що поєднав у собі приводний електричний двигун і індустріальний механічний редуктор. Його використання значно спростило розробку та виготовлення машини, її надійність. Мотор служить для перетворення електричної енергії в механічну.

На рисунку 4.1 показано електроциліндр: в розрізі (а) і в зборі (б). Електроциліндр складається з електродвигуна 1, гвинта 2 (одночасно є ротором двигуна) та нерухокої гайки 3. Дві останні частини утворюють кульогвинтову передачу, в якій обертання гвинта призводить до його поступального руху.



а)



б)

Рисунок 4.1 – Електроциліндр: а – у розрізі; б – у зборі

На рисунку 4.2 зображено мотор-колесо. Канадські конструктори створили уніфікований колісний модуль, що складається з власне колеса, електричного мотора, гальмівної системи, підвіски і контролера, енергопостачання якого здійснюється від блоку батарей автомобіля.

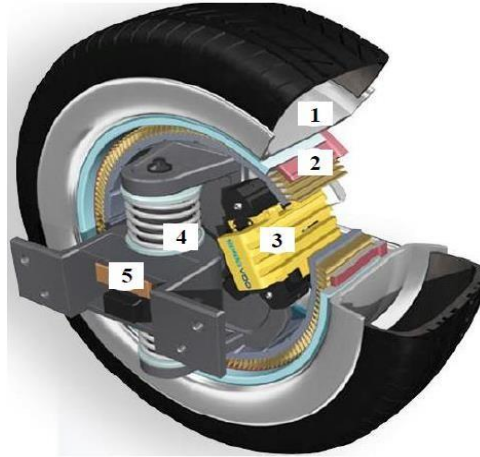


Рисунок 4.2 – Мотор-колесо: 1 – обід колеса; 2 – електродвигун; 3 - механізм гальмування; 4 – підвіска; 5 – електрорульове керування

Використання таких мотор-колів позбавляє електромобільні «стартапи» від необхідності розробляти кожен із компонентів самостійно: вони просто під'єднують уніфіковані блоки до рами свого транспортного засобу.

А мотор-колесо, наприклад типу Protean 360, дає змогу електромобілям паркуватися будь яким боком.

Команди розгону, маневрування і гальмування будуть передаватися в модулі не механічно, а за допомогою електроніки. Перевага такої системи ще й в тому, що вона дозволяє відмовитися від рульової колонки, а це робить просторішим салон пасажирського або вантажного електромобіля.

На рисунку 4.3 зображено верстат із числовим програмним керуванням з позначеним електрошпинделем.



Рисунок 4.3 – Верстат з позначенням електрошпинделя

Поширеним у складі верстатів, як модуль руху, є електрошпindel. Він являє собою високошвидкісний електричний двигун, робочий вал якого оснащений спеціальним пристосуванням, призначеним для закріплення оброблюючої заготовки чи різального інструменту. Шпindel для верстата оснащується спеціалізованими підшипниками зі сталі чи кераміки, що дозволяють досягати за допомогою перетворювача частоти швидкості обертання валу двигуна від 3000 об/хв до 24000 об/хв (стандарт), а деяких випадках - до 60000 об/хв. Ця властивість дозволяє здійснювати шпindelом обробку всіх можливих матеріалів: дерево, метал, алюміній, скло, камінь, пластик.

4.2 Мехатронні модулі руху

Мехатронний модуль руху (ММР) на відміну від модуля руху додатково включає ще електротехнічну, електронну й інформаційну частини. Вбудовані в модулі руху мініатюрні датчики й електронні блоки для обробки їх сигналів перетворюють МР у мехатронні модулі руху (ММР). Для створення сучасних автоматичних ліній необхідні різноманітні мехатронні модулі руху, що задовольняють ряд вимог: висока точність реалізації рухів, надійність, довговічність, можливість роботи за наявності різних видів збурень і в широкому діапазоні температури навколишнього середовища, а також значно менші масогабаритні показники у порівнянні зі 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.

Належної компактності модуля руху можна досягти 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.

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.

4.3 Інтелектуальні модулі руху

Інтелектуальний мехатронний модуль (ІММ) – конструктивно і функціонально самостійний виріб із інтеграцією механічної (гідравлічної,

пневматичної), електротехнічної й комп'ютерної (мікропроцесорної) частин, який можна використовувати індивідуально і в різних комбінаціях з іншими модулями.

Таким чином, у порівнянні із ММР, у конструкцію ІММ додатково включені мікропроцесорне обчислювальне обладнання і силові електронні перетворювачі, а також елементи, що забезпечують інтелектуальне керування.

Розглянемо основні переваги, які дає застосування інтелектуальних мехатронних модулів:

1. здатність ІММ виконувати складні рухи самостійно, без „звертання” до верхнього рівня керування, що підвищує автономність модулів, гнучкість і живучість мехатронних систем;

2. спрощення комунікацій між модулями й центральною системою керування (наприклад, перехід до безпроводних комунікацій);

3. підвищення надійності й безпеки мехатронних систем завдяки комп'ютерній діагностиці несправностей і автоматичному захисту в аварійних і позаштатних режимах роботи;

4. 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;

5. інтелектуалізація сенсорів для мехатронних модулів, що дозволяє досягти більш високої точності вимірювання, програмним шляхом забезпечивши в самому сенсорному модулі фільтрацію шумів, калібрування, лінеаризацію характеристик «вхід-вихід», компенсацію гістерезису й дрейфу нуля.

Інтелектуальний мехатронний модуль складається з наступних основних елементів:

1. електродвигуна (хоча можливе використання приводів і інших типів, наприклад, гідравлічних);

2. механічного перетворювача;

3. інформаційних пристроїв (датчиків зворотного зв'язку й сенсорного обладнання);

4. пристрою керування (контролера).

Мехатронні модулі різного рівня наведені на рисунку 4.4

У якості механічних перетворювачів руху застосовують зубчасті, гвинтові та інші передачі.

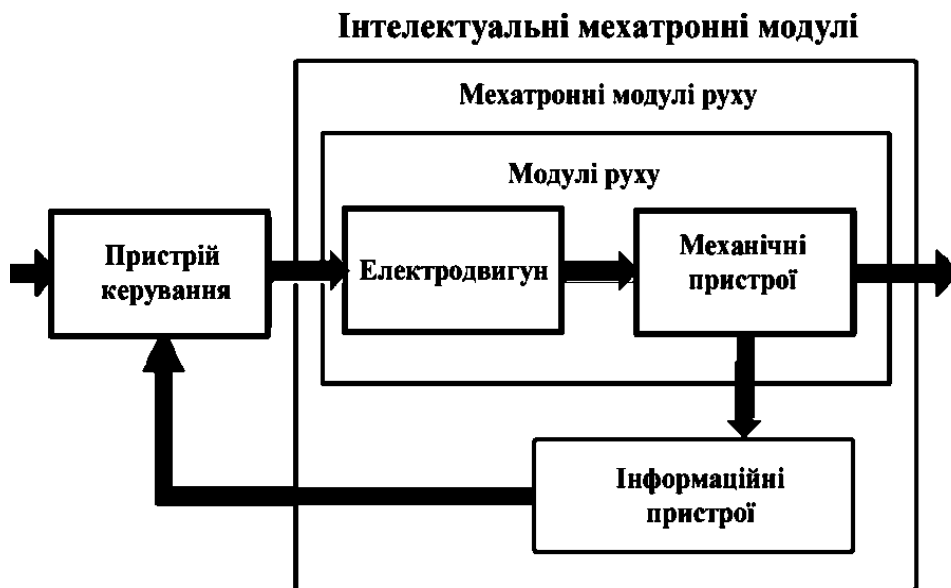


Рисунок 4.4 – Структура інтелектуального мехатронного модуля

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. У ланці мехатронної техніки одне з перших місць займає інтелектуальна мехатронна машина. *Інтелектуальна мехатронна машина (ИММ)* – це інтелектуальна багатовимірна система, побудована на мехатронних принципах і технологіях. Така система здатна ефективно виконувати програми функціональних рухів в умовах нечіткої та неповної інформації про експлуатаційну характеристику машини та параметри зовнішнього середовища. На рисунку 4.5 зображено узагальнену структуру мехатронної машини, в основу побудови якої покладена структура автоматичних роботів. Зовнішнім середовищем для машин цього класу є технологічне середовище, яке містить основне й допоміжне обладнання, технологічне оснащення та об'єкти робіт. Під час виконання мехатронною системою заданого функціонального руху, об'єкти робіт впливають на робочий орган. Наприклад, вплив сили різання на технологічну операцію механічної обробки, вплив реакції струменя рідини під час гідравлічного різання та інші.

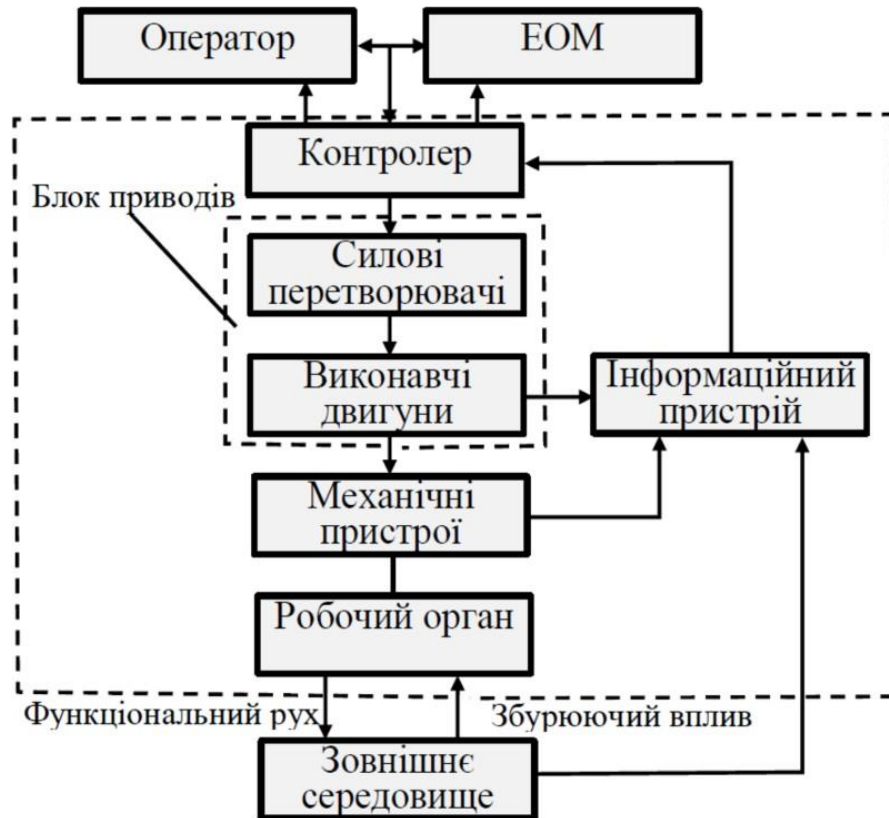


Рисунок 4.5 – Узагальнена структурна схема інтелектуальної мехатронної машини

До складу мехатронної машини входять чотири частини:

1. механічний пристрій, кінцевою ланкою якого є робочий орган;
2. блок приводів, що складається з силових перетворювачів і виконавчих двигунів;

3. пристрій комп'ютерного керування, на вхід якого надходять команди людини-оператора або ЕОМ верхнього рівня керування;

4. інформаційний пристрій, призначений для отримання й передачі до пристрою комп'ютерного керування даних про реальний рух машини та про фактичний стан його підсистем.

Механічний пристрій і двигуни об'єднані в групу виконавчих пристроїв. До складу групи інтелектуальних пристроїв включені електронна, керувальна та інформаційна частини машини.

Механічний пристрій мехатронної машини є багатоланковим механізмом. Кінематичний ланцюг його утворюють рухомі ланки – складові кінематичні пари. Кінцевою ланкою кінематичного ланцюга є робочий орган.

Робочий орган мехатронної машини – це складова частина механічного пристрою, що призначений для безпосереднього виконання технологічних операцій.

Прикладами робочих органів в робототехніці є такі пристрої як: механічні, вакуумні й електромагнітні захватні, а також зварювальні кліщі (для точкового зварювання), інструментальні головки для механічної обробки й лазерних операцій, фарбувальний пістолет тощо. Отже, робочий орган – це керований модуль, який може мати декілька ступенів рухливості та складатися з декількох елементів.

Пристроєм комп'ютерного керування є комплекс апаратних і програмних засобів, що виробляє сигнали керування для блоку приводів машини.

До складу комплексу входять мікропроцесорний контролер, задаючі пристрої (наприклад, джойстики та рукоятки), пульт керування оператора, обчислювальні та перетворювальні пристрої, периферійні пристрої введення-виведення інформації.

Пристрій комп'ютерного керування виконує такі основні функції:

1. керування функціональними рухами мехатронної машини у реальному масштабі часу;
2. координації керування механічним рухом із супутніми зовнішніми процесами;
3. взаємодії з людиною-оператором через людино-машинний інтерфейс у режимах програмування та безпосередньо у процесі руху (режим on-line);
4. обміну даними із зовнішніми пристроями (інформаційним пристроєм, блоком приводів, комп'ютером верхнього рівня, периферійними пристроями).

Сучасні інтелектуальні модулі (рис. 4.6), вузли та системи легко перепрограмуються під нове завдання, що розширює функціональні можливості машин і механізмів.



Рисунок 4.6 – Мехатронні модулі з інтегрованим робочим органом

Водночас із розвитком техніки вузли машини різного фізичного походження (механічні, електричні, електромеханічні, електронні, інформаційні) поступово

об'єднуються в єдине конструктивне ціле. Саме такі інтелектуальні машини та вузли називають *мехатронними*.

Інформаційний пристрій призначений для збору й передачі в пристрій керування інформації про фактичний стан зовнішнього середовища й руху мехатронної машини.

В інформаційному пристрої можна виділити такі групи сенсорів:

1. давачі інформації про стан зовнішнього середовища й об'єктів робіт (системи технічного зору, локаційні давачі, далекоміри тощо);
2. давачі інформації про рух механічної частини (давачі переміщень, швидкостей, прискорень, сил і моментів);
3. давачі зворотного зв'язку блоку приводів (надають інформацію про поточні значення електричних струмів і напруг у силових перетворювачах).

Система інтелектуального керування машиною (комплекс верхнього рівня керування) в умовах неповної інформації зазвичай реалізується у вигляді комплексу програмних засобів на комп'ютері верхнього рівня керування.

ЗАВДАННЯ ДЛЯ ПЕРЕВІРКИ ЗНАНЬ

1. Що таке робот?
2. Класифікація промислових роботів за їх призначенням
3. Застосування технологічних промислових роботів
4. За скількома ступенями рухливості можна програмувати маніпулятор робота?
5. Складові промислового робота
6. Основні параметри роботів, що визначають їх динамічні властивості
7. Швидкодія маніпулятора визначається яким параметром?
8. Діапазони швидкодії маніпуляторів у роботів загального застосування
9. Точність маніпулятора і системи пересування робота характеризується яким показником?
10. Найчастіше точність роботів характеризують яким показником?
11. Класифікація мехатронних об'єктів залежно від рівня інтеграції компонентів і ступеня складності
12. Види класифікацій мехатронних об'єктів
13. У яких пристроях знаходять широке застосування мехатронні модулі і системи?
14. Що слугує методологічною основою розробки мехатронних систем?
15. Вкажіть основні вимоги, щодо будови мехатронних модулів руху для багатьох областей їх застосування

16. Відмінність мехатронного модуля руху від модуля руху
17. Які вимоги необхідні для створення сучасних автоматичних ліній з різноманітними мехатронними модулями руху?
18. Відмінність конструкції інтелектуального мехатронного модуля від мехатронного модуля руху
19. Складові інтелектуального мехатронного модуля

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Гуржій А. М., Нельга А. Т., Співак В. М., Ітякін О. С. Основи автоматики та робототехніки: навч. посіб. Дніпро: «Гаран СВ», 2021. 243 с. URL: http://ela.kpi.ua/bitstream/OAR_NT (дата звернення: 19.10.2023)
2. Артюх О. М., Дударенко О. В., Кузьмін В. В., Сосик А. Ю., Щербина А. В. Основи мехатроніки: навч. посіб. Запоріжжя: НУ «Запорізька політехніка», 2021. 372 с. URL: http://eir.zp.edu.ua/bitstream/123456789/8209/1/NP_Artyukh.pdf. (дата звернення: 19.10.2023)
3. Крак Ю. В. Основи теорії керування та робототехніки: навч. посіб. для студентів спеціальності «Інформатика». Київ: Видавничо-поліграфічний центр «Київський університет», 2021. 152 с. URL: https://csc.knu.ua/media/filer_public/9b/3b/9b3bdba9-15dc-4c50-ad54-2bb54713b580/osnovi_teoriyi_keruvannia_ta_robototekhniki.pdf (дата звернення: 19.10.2023)
4. Сучасні електромехатронні комплекси і системи: навч. посібник / Т. П. Павленко, В. М. Шавкун, О. С. Козлова, Н. П. Лукашова; Харків. нац. ун-т міськ. госп-ва ім. О. М. Бекетова. Харків: ХНУМГ ім. О. М. Бекетова, 2019. 117 с. URL: https://eprints.kname.edu.ua/55311/1/2019%20%D0%BF%D0%B5%D1%87.%20%2047%D0%9D%20%D0%9F%D0%BE%D1%81.%20%D0%A1%D0%95%D0%9C%D0%A1_%28%D0%9A%D0%BE%D0%B7%D0%BB%D0%BE%D0%B2%D0%B0%29_03.10.pdf (дата звернення: 03.02.2024)
5. Паламар М. І., Стрембіцький М. О. Системи прецизійного управління мехатронних систем: конспект лекцій для студентів спеціальностей 175 «Інформаційно-вимірювальні технології», 176 «Мікро- та наносистемна техніка». Тернопіль: ТНТУ ім. Івана Пулюя, 2023. 94 с. URL: https://elartu.tntu.edu.ua/bitstream/lib/42848/1/%d0%9a%d0%be%d0%bd%d1%81%d0%bf%d0%b5%d0%ba%d1%82%20%d0%bb%d0%b5%d0%ba%d1%86%d1%96%d0%b9%20%d0%a1%d0%9f%d0%a3%d0%9c%d0%a1_2023_1.pdf (дата звернення: 21.02.2024)

Навчально-методичне видання

АВТОМАТИЗАЦІЯ ТА КОМП'ЮТЕРНО-ІНТЕГРОВАНІ ТЕХНОЛОГІЇ

Магістерський курс

Навчальний посібник

**Гуменюк Л. О., Гуменюк П. О., Пальчевський Б. О.,
Решетило О.М., Сацик В.О., Смолянкін О. О., Федік Л. Ю.**

Загальна редакція к.т.н., доцента Гуменюка П. О.

Комп'ютерний набір: колектив авторів
Комп'ютерна верстка: Л.О. Гуменюк

Друкується в авторській редакції

Підп. до друку «__» ____ 2024 р. Папір офс.
Гарнітура Таймс. Ум. друк. арк. 12,8. Обл.-вид. арк. 1.
Тираж 100 прим.

Відділ іміджу та промоції
Луцького національного технічного університету
43018 м. Луцьк, вул. Львівська, 75
Друк – ВІП ЛНТУ

