

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Київський національний університет будівництва і архітектури

ЛОГІСТИКА МЕХАТРОННИХ СИСТЕМ БУДІВЕЛЬНОЇ ТЕХНІКИ

Конспект лекцій
для здобувачів першого (бакалаврського) рівня вищої освіти
спеціальності 131 «Прикладна механіка»

Київ 2024

УДК 69.002.5

M21

Рецензент К.І. Почка, д-р техн. наук, професор

Затверджено на засіданні навчально-методичної ради Київського національного університету будівництва і архітектури, протокол № 9 від 25 квітня 2024 року.

Міщук Д.О.

M21 Логістика мехатронних систем будівельної техніки: конспект лекцій / Д.О. Міщук, В.О. Волянчук, С.Ю. Комоцька. – Київ : КНУБА, 2024. – 176 с.

Подано інформацію про мехатронні системи та їхнє застосування в будівельній техніці. Наведено принципи їхньої будови, компоненти мехатронних систем, компоненти інформаційних систем, застосовуваних в мехатроніці, та основи їхнього розрахунку.

Рекомендовано для здобувачів першого (бакалаврського) рівня вищої освіти спеціальності 161 «Прикладна механіка».

УДК 69.002.5

© Д.О. Міщук, В.О. Волянчук, С.Ю. Комоцька, 2024

© КНУБА, 2024

ЗМІСТ

Вступ.....	4
Змістовий модуль 1. Теоретичні основи мехатронних систем.....	5
Лекція 1. Загальні відомості про мехатронні системи будівельної техніки.....	5
Лекція 2. Моделі мехатронних модулів будівельної техніки.....	23
Лекція 3. Принципи побудови мехатронних структур.....	48
Лекція 4. Сучасні методи управління мехатронними модулями.....	67
Лекція 5. Пристрої мехатронних систем в логістиці.....	95
Змістовний модуль 2. Логістика мехатронних систем.....	113
Лекція 6. Основні принципи SCM-логістики.....	113
Лекція 7. Процеси логістики будівельної техніки.....	126
Лекція 8. Логістика мехатронної системи.....	137
Лекція 9. Програмування мехатронних модулів.....	148
Список літератури.....	175

Вступ

Логістика мехатронних систем – це напрям науки, у якому поєднано принципи логістики та мехатроніки, що застосовуються у створенні, оптимізації управління і функціонуванні складних технічних систем з автоматизованим управлінням і ознаками «смарт»-технологій. У світі, де технології швидко розвиваються і потребують більш ефективного управління, розуміння принципів розумного управління є надзвичайно важливим для фахівців різних галузей, а мехатронні системи є основними компонентами для побудови «розумних» пристроїв та створення сучасних автоматизованих машин.

У процесі вивчення курсу студенти повинні засвоїти основні поняття та визначення мехатроніки, а також найголовніші елементи мехатронних систем та їхні взаємозв'язки з процесами функціонування машини.

Мета курсу – опанування принципів, стратегій і методів створення мехатронних систем для управління різними логістичними потоками.

За результатами вивчення курсу здобувачі мають:

- знати:

- конструкції та принципи роботи мехатронних систем;
- правила побудови і експлуатації мехатронних модулів руху;
- сучасні технологічні та інформаційні засоби, які використовуються для підтримки логістики мехатронних систем, такі як системи автоматизованого управління та принципи програмування мехатронних модулів;

- вміти:

- аналізувати конструкції та режими роботи мехатронних систем;
- розробляти рекомендації до запровадження мехатронних модулів руху.

Всі компоненти курсу викладено відповідно до сучасних тенденцій розвитку технологій та практичних вимог сучасних підприємств до застосування мехатронних систем.

Конспект лекцій призначений також для тих, хто цікавиться принципами функціонування, будови та логістики мехатронних систем і бажає розширити свої знання в цій сфері.

Змістовий модуль 1. ТЕОРЕТИЧНІ ОСНОВИ МЕХАТРОННИХ СИСТЕМ

Лекція 1. Загальні відомості про мехатронні системи будівельної техніки

Мехатронні системи – це технічні системи, що забезпечують виробничі процеси, основані на комплексному використанні електричних, механічних та інформаційних технологій. Вперше термін «мехатроніка» запропонував старший інженер Yaskawa Electric Тецуро Морія в 1969 році. Термін складається з двох частин – «меха», від слова механіка і «троніка», від слова електроніка [1]. В розумінні цього терміна йдеться про поєднання трьох галузей науки:

- електроніки, складовими якої є мікроелектроніка, силова електроніка, електричні перетворювачі та вимірювачі інформації;
- механіки і електромеханіки, що розглядає механічні елементи, машини, приводи, точну механіку;
- інформаційні технології, що охоплює теорію систем автоматичного управління, алгоритмічне моделювання, програмне забезпечення, штучний інтелект.

Мехатроніка – це галузь науки міждисциплінарного характеру, котра ґрунтується на таких галузях, як (рис. 1.1) механіка (теоретична механіка, машинобудування, теорія машин і механізмів), електротехніка (електроніка, мікропроцесорна техніка, силова та вимірювальна електротехніка, електропривод), інформатика (теорія систем, обробка сигналів, штучний інтелект).

Метою мехатроніки є створення нових або вдосконалення вже наявних автоматично або програмно керованих модулів руху, на основі яких можна створювати рухомі інтелектуальні машини та системи. В мехатроніці важливий інтерес становлять задачі із створення навантаження на машину, за наявності перешкод за заданого закону руху її механізмів або синтезу системи приводу й управління за зворотного розгляду проблеми. Технічне розв'язання таких завдань складається з етапів вимірювання, регулювання, установлення, проєктування та інтегрування приводів, датчиків, системи управління й обробка інформації.



Рис. 1.1. Структура дисципліни мехатроніки

Предметом дисципліни мехатроніка є процеси проектування та виробництва модулів машин і технічних систем для дотримання програмованих функціональних режимів руху. Під функціональним програмованим режимом руху розуміють кероване переміщення виконавчого механізму механічної системи, яке координується з паралельно керованими технологічними та інформаційними процесами.

Базовими об'єктами мехатронних систем є мехатронні модулі різного призначення, тому проектування сучасних мехатронних систем ґрунтується на модульних принципах і технологіях.

На рис. 1.2 наведено загальну схему потоків енергії в машині під час виконання технологічного процесу з позиції мехатронної системи.

Згідно з рис. 1.2 потік енергії надходить до машини через деякий перетворювач, а потім або безпосередньо використовується споживачем енергії, або перетворюється в іншу форму енергії. Форма енергії може бути електрична, механічна (потенційна або кінетична, гідравлічна, пневматична), хімічна або термічна. Потік енергії в такій системі може бути безперервним або періодичним (повторюваним) потоком, хоча для деяких механічних процесів, таких як механічні передачі або прецизійні механічні пристрої, характерними є кускові, або переривчасті, потоки енергії. Елементами механічної частини можуть бути робочий орган, який виконує корисну механічну роботу (переміщення вантажу, механічну обробку деталі тощо) або механічна передача, яка змінює швидкість руху чи його характер (поступальний замість обертового).

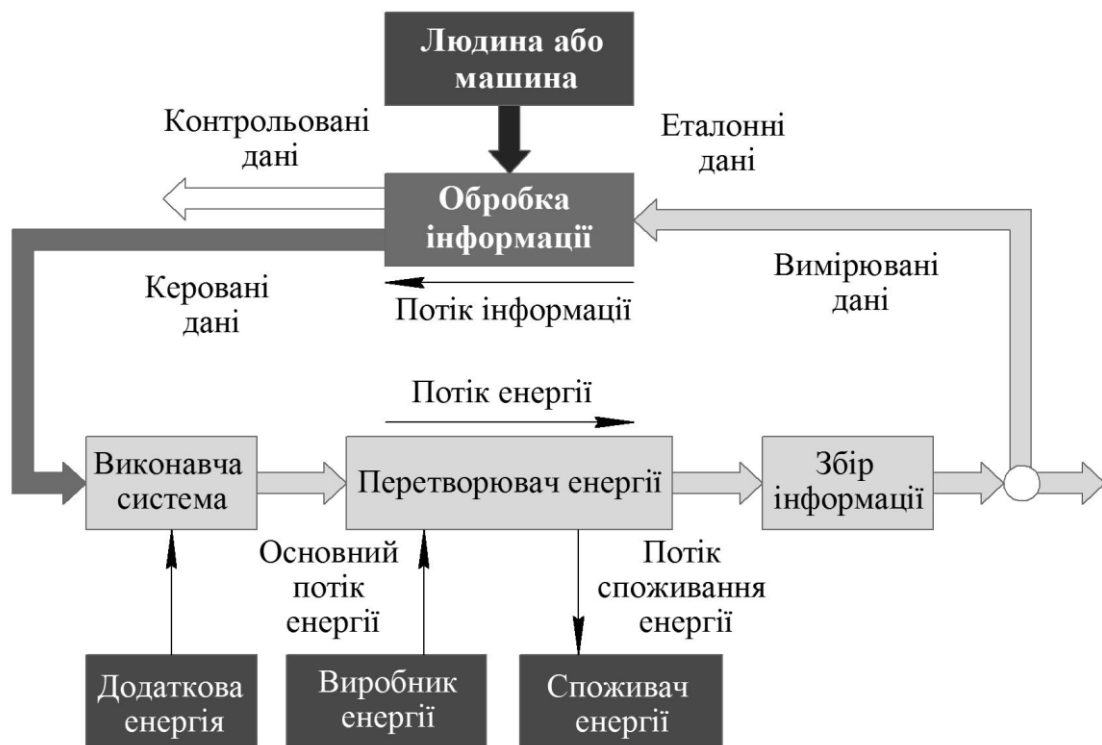


Рис. 1.2. Схема розподілу та контролювання енергетичних потоків в машині

Головним елементом приводної частини є привод, який являє собою автоматизовану, зазвичай електрифіковану систему, що складається з електричних, механічних та електронних вузлів. Система керування належить до складу перетворювача енергії, вона містить автоматизовану систему, що керує процесом стабілізації швидкості двигуна. Двигун – перетворювач енергії, що може бути електричним, гідравлічним, пневматичним або двигуном внутрішнього згоряння. Найбільшого застосування в мехатронних модулях набули електричні двигуни.

Потік енергії звичайно є продуктом узагальненого потоку та потенціалу (зусилля) під час виконання машиною різноманітних технологічних процесів. Інформацію про стан процесу в механічних системах можна отримати, вимірюючи узагальнені потоки (швидкість, об'ємні або масові витрати), електричний струм або потенціал (силу, тиск, температуру або напругу). Разом з еталонними змінними вимірювані змінні є вхідними даними для обробника потоку інформації через цифрову електроніку, де відбувається визначення похибки між заданими даними користувача та контрольованими даними. Передавання даних до виконавчої системи згідно із запропонованою

схемою відбувається з урахуванням похибки, вимірної та обробленої інформаційною системою. Саме такий підхід до синтезу та управління технічними системами і покладено в основу мехатроніки.

Виокремлюють такі мехатронні системи:

- мехатронні модулі;
- мехатронні машини;
- мехатронні транспортні засоби;
- прецизійні мехатронні системи;
- мікромехатроніка.

В мехатроніці застосовують системний підхід. Наприклад, будь-яка машина являє собою складну систему, яка поділяється на механічну та електричну підсистеми (електромеханічна, електрогідравлічна, електропневматична, комбінована), систему управління та інформаційну підсистеми. Зауважимо, що інформаційну підсистему виділяють, як окрему, для інтелектуальних мехатронних систем. У примітивних мехатронних модулів інформаційної системи може не бути.

Застосування замкнутого циклу керування в мехатронних системах для контролю швидкості або сили не лише сприяє точному виконанню машиною еталонних параметрів, але й наближує роботу такої машини до лінійної, навіть якщо механічні системи мають нелінійних характер управління.

Сучасні мехатронні системи дають змогу виконувати складні функції, які неможливо виконати без цифрової електроніки, тому електроніка в мехатронних системах значною мірою інтегрована в традиційні механічні системи. Такий підхід дозволяє контролювати параметри машини, які не підлягають прямому вимірюванню, шляхом вимірювання непрямих опосередкованих сигналів і впливати на основні параметри за допомогою прямого або зворотного зв'язку. Наприклад, такі параметри, як ковзання шин транспортного засобу, швидкість руху транспортними засобами, або такі параметри, як амортизація, коефіцієнти жорсткості та демпфування, вимірюють та контролюють за результатами вимірювання переміщень або прискорень. Іншим прикладом застосування непрямого контролю є інтегрований нагляд та діагностика несправності машини, що може відбуватися дистанційно на підставі даних з датчиків, які формуються та накопичуються в електронній системі, завдяки цьому система стає більш «інтелектуальною».

Інтеграція в межах мехатронної системи може відбуватися через інтеграцію компонентів і через інтеграцію обробки інформації. Інтеграції компонентів (апаратна інтеграція) досягають, вбудовуючи датчики і мікрокомп'ютери у виконавчі механізми механічної системи (рис. 1.3).

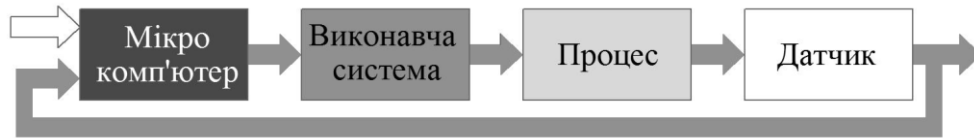


Рис. 1.3. Принципова схема апаратної інтеграції

Завдяки інтеграції мікроконтролерів із сенсорами утворюються інтелектуальні датчики, а інтеграція приводів та мікрокомп'ютерів сприяє створенню інтелектуальних виконавчих механізмів.

Інтеграція на рівні обробки інформації (інтеграція програмного забезпечення) ґрунтується на розширенні функцій керування механізмом. Наприклад, крім основного контролю прямого та зворотного зв'язку, додатковий вплив може мати знання процесу та відповідна обробка поточної інформації (рис. 1.4).

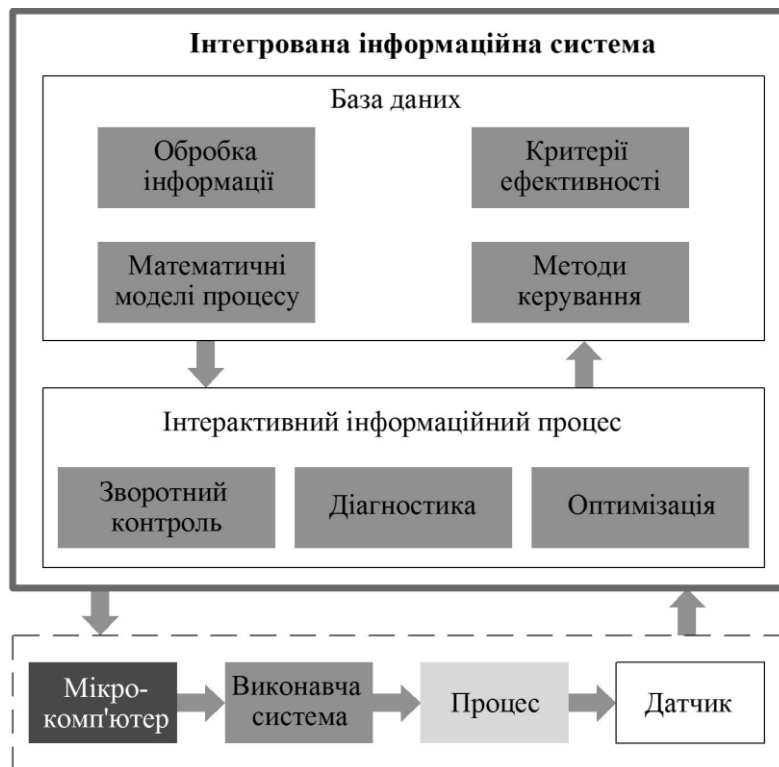


Рис. 1.4. Принципова схема інформаційної інтеграції

Завдяки такому підходу відбувається обробка доступних сигналів на вищих рівнях системи, зокрема інтерактивне спостереження та діагностика, оптимізація та загальне управління процесом. Це дає змогу скористатися алгоритмами реального часу, які повинні бути адаптовані до механічних властивостей процесу, виражених математичними моделями у формі статичних характеристик або диференціальних рівнянь. База знань містить методи керування, моделі процесу та критерії ефективності.

Отже, керувати механічною системою можна різними способами завдяки обробці інформації більш високого рівня, позначеної інтелектуальними властивостями, ймовірно, з можливостями навчання, таким чином формуючи інтеграцію за допомогою програмного забезпечення, адаптованого до процесу.

В мехатронних системах обробка даних може відбуватися на декількох рівнях (рис. 1.5):

- *рівень 1*: рівень процесу (найнижчий рівень);
- *рівень 2*: рівень регулювання та зворотного зв'язку на рівні процесу;
- *рівень 3*: рівень спостереження (контроль граничних значень) та діагностика помилок, застосування простих заходів для подальшої роботи чи зупинки;
- *рівень 4*: рівень загального керування та координації окремих складових системи й оптимізації процесів.

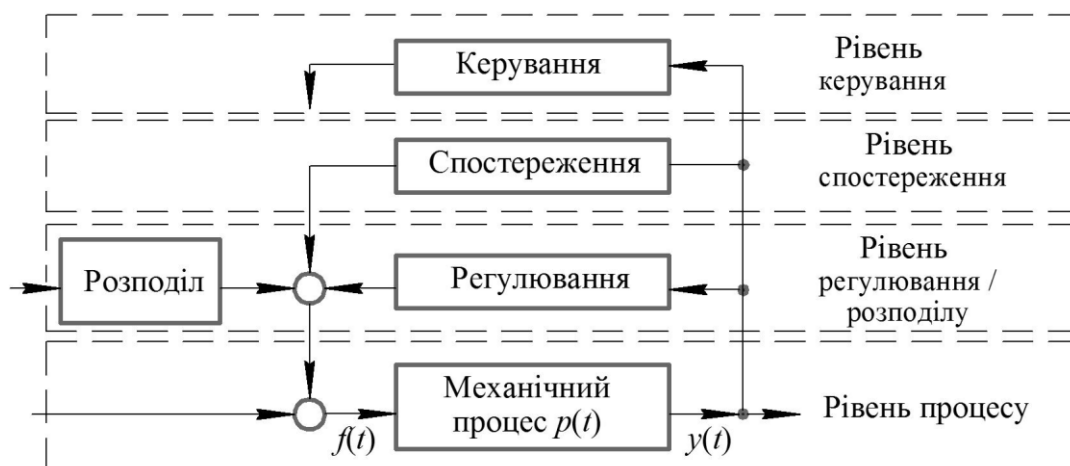


Рис. 1.5. Рівні обробки даних процесу

Обробка інформації та регулювання мехатронними системами сигналів на нижніх рівнях виконують із застосуванням простих алгоритмів та програмних модулів в умовах реального часу, тобто відтворюються функції керування, регулювання і простого контролю. Обробка інформації на нижчих рівнях відбувається за алгоритмами, які містять вільно налаштовуванні параметри і повинні бути адаптовані до статичного та динамічного перебігу процесу. На відміну від ручного налаштування методом спроб і помилок використання математичних моделей забезпечує точну та швидку автоматичну адаптацію. Математичні моделі можна отримати шляхом ідентифікації та оцінки параметрів, використовуючи виміряні та вибірккові вхідні та вихідні сигнали. Застосування цих методів не обмежується лінійними моделями, що дає змогу використовувати декілька класів нелінійних систем.

Якщо методи оцінювання параметрів поєднуються з відповідними методами синтезу алгоритму керування, то виникають адаптивні системи керування. Цифрова обробка сигналів у поєднанні з алгоритмами синтезу програм керування дають змогу створювати розумні інтелектуальні системи (системи зі штучним інтелектом).

До мехатронних систем ставлять ряд вимог, зокрема такі:

- властивість якісно виконувати нові службові та функціональні завдання;
- досягнення надвисоких швидкостей руху кінцевої ланки;
- надвисока точність позиціонування модулів руху;
- компактність модулів руху та систем відтворення руху;
- властивості інтелектуальної поведінки систем, які функціонують в невизначених робочих середовищах;
- виконання просторових переміщень по криволінійних траєкторіях з дотриманням складних законів руху.

Збільшення кількості автоматичних функцій (автономії), зокрема електронних компонент, датчиків та виконавчих механізмів, призводить до підвищення вимог до надійності та безпеки, тому для мехатронних систем важливою стає комплексна діагностика несправності. На рис. 1.6 відображено процес, на який впливають несправності системи, що призводять до відхилення від нормальних умов її роботи і можуть бути згенерованими як ззовні, так і зсередини.

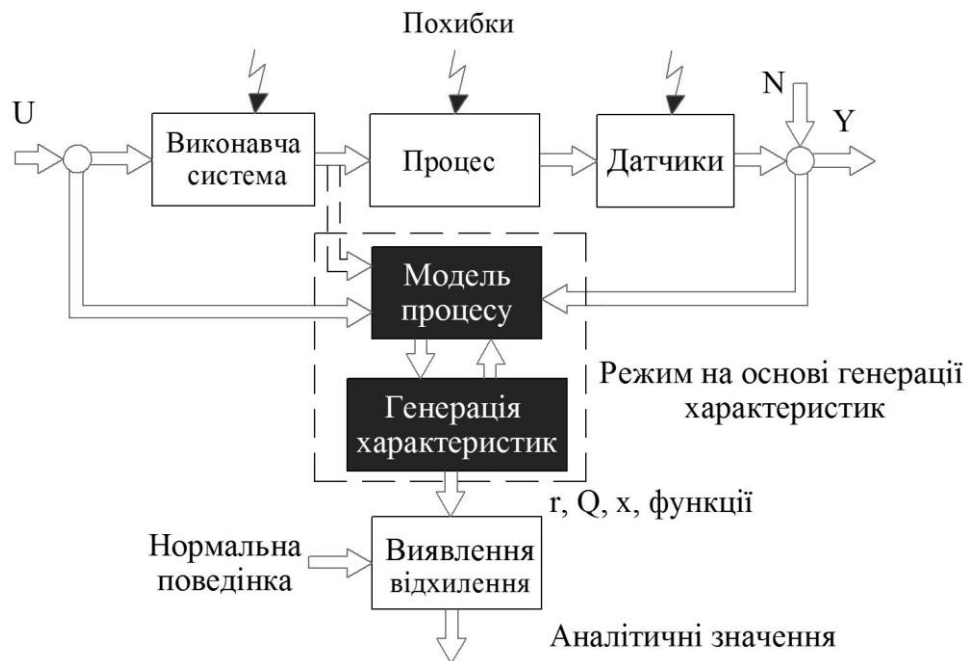


Рис. 1.6. Схема процесу з накладанням похибок

Зовнішні несправності можуть бути спричинені джерелом живлення, забрудненням або зіткненням, внутрішні несправності – спрацьованістю, незмащеністю або несправністю приводу чи датчика. Класичним способом виявлення несправності є перевірка граничного значення кількох вимірюваних змінних. Однак початкові та періодичні несправності зазвичай не можуть бути виявлені, а поглиблена діагностика несправностей є неможливою в разі цього простого підходу. Методи виявлення несправностей на основі моделей та методів діагностики, які були розроблені останніми роками, дають змогу виявляти невеликі ранні несправності за допомогою вимірюваних вхідних $U(t)$ і вихідних сигналів $Y(t)$ та моделей процесів, які генерують параметри оцінки стану системи. Отримані значення порівнюють із параметрами системи для її нормальної поведінки, а за допомогою методів класифікації виконують якісну діагностику несправності.

Зазначимо, що одна й та сама модель процесу може бути використана і для конструкції контролера, і для виявлення його несправності. Загалом моделі безперервного часу є найкращими для виявлення несправності, проте застосовують і моделі з дискретним часом.

Мехатронні системи широко застосовують в робототехніці, верстатобудуванні, автомобілях, галузевому машинобудуванні, логістиці (рис. 1.7).



Рис. 1.7. Приклади застосування мехатронних систем

Мехатронні системи будівельної техніки – це автоматизовані системи, які застосовують для автоматизації роботи машин та їхніх механізмів і які можуть бути застосовані в процесах зведення інженерних об’єктів і споруд. Застосування мехатронних систем в будівництві дає змогу знизити витрати на робочу силу та підвищити продуктивність втілення багатьох типів проєктів. Мехатронні системи будівельної техніки використовують для виконання різних видів роботи, наприклад для переміщення вантажів, контролю подавання і дозування різноманітних матеріалів тощо.

Мехатронні системи будівельної техніки складаються з багатьох компонентів – двигунів, датчиків, програмувальних пристроїв та елементів, які можуть бути складені в автоматизовану систему управління. У базовій структурі компоненти мехатронної системи можна подати у виді структурної схеми (рис. 1.8), у складі якої є механічна система, виконавчий механізм, датчики, системи обробки даних та процеси (верхній рівень – взаємодія потоків енергії та нижній рівень – потік даних). Згідно з такою схемою, енергія, що надходить до робочих органів машини, перетворюється на силу або крутний момент, які виконують корисну роботу механічної системи. Датчики фіксують виконану роботу та передають інформацію до системи обробки даних, яка аналізує та подає сигнали до системи керування робочими органами

машини. Таким чином відбувається керування потоками енергетичних процесів в механічній системі машини. Усі ці процеси можна описати як кінематичні, динамічні або теплові функції.

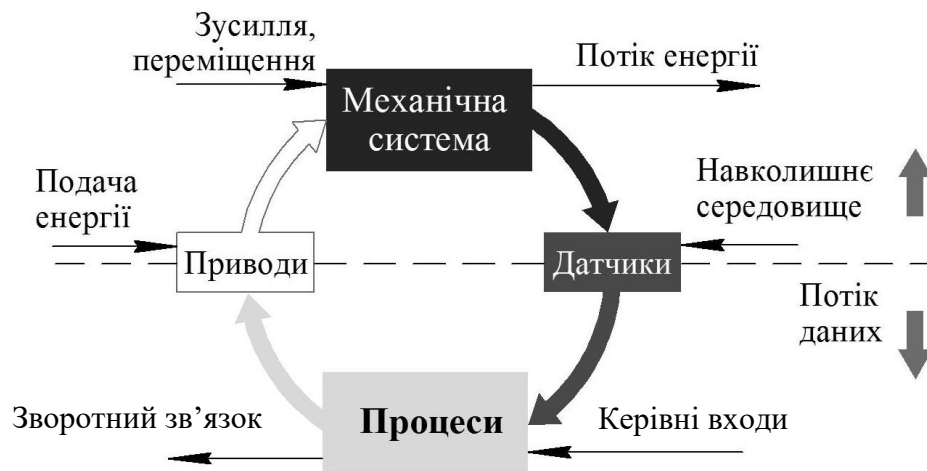


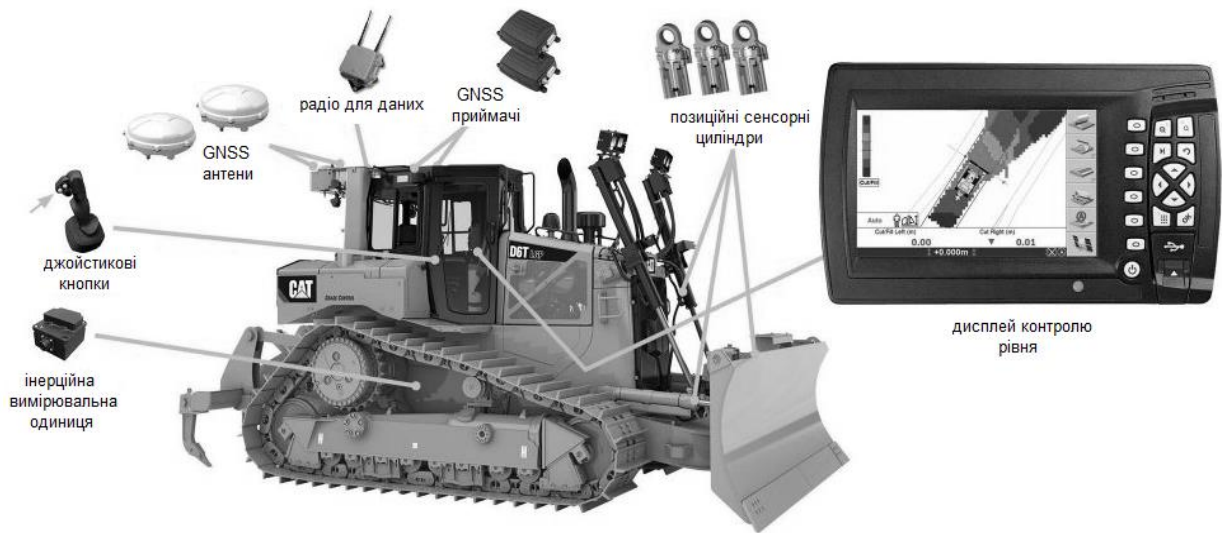
Рис. 1.8. Структурна схема мехатронної системи будівельної машини

Розглянемо декілька прикладів застосування мехатронних систем в будівельних машинах.

На рис. 1.9 показано принципову схему мехатронної системи бульдозера на прикладі машини від компанії Caterpillar. Бульдозери багатьох виробників, як і компанії Caterpillar зокрема, мають схожі інтегровані електронні системи керування двигуном та трансмісією, які покращують характеристики машин завдяки високому ступеню інтеграції компонентів силової установки. Це означає, що двигун та трансмісія можуть взаємодіяти між собою більш ефективно, що суттєво впливає і на інші функції бульдозера. Окрім того, у бульдозерах Caterpillar використовують систему Traction Control (система контролю тяги), принцип дії якої полягає в автоматичному зниженні обертів двигуна у разі пробуксовування гусениць.

У машинах Caterpillar встановлено електронну систему керування, яка складається з бортового комп'ютера та електронних органів керування з аналоговими або цифровими датчиками (залежно від моделі машини). Крім того, в бульдозерах компанії Caterpillar може бути встановлена дистанційна система керування Command, яка уможливорює нові способи віддаленого керування деякими бульдозерами та екскаваторами. Завдяки повній інтеграції Command з електронною та гідравлічною системами машини дистанційне керування забезпечує

швидкий відгук, плавне керування. Існують системи з режимами прямої видимості (LOS) або непрямой (NLOS) з діапазоном візуального контакту до 400 м.



а



б

Рис. 1.9. Розміщення елементів системи автоматичного управління та контролю бульдозера (*а*) та вигляд робочого місця машиніста з електронним керуванням (*б*)

На рис. 1.10 показано архітектуру мехатронної системи колісного трактора від компанії John Deere, до складу якої входить термальна камера та набір візуальних камер, GPS-навігатор, лідар, колісний енкодер, бортовий комп'ютер. Така система дає змогу виконувати візуальну ідентифікацію об'єктів та параметрів зовнішнього середовища, оброблювати їх та реагувати на їхні зміни. Застосування декількох дублювальних систем візуального нагляду дає можливість резервувати або об'єднувати дані з датчиків, підвищуючи їхню надійність.

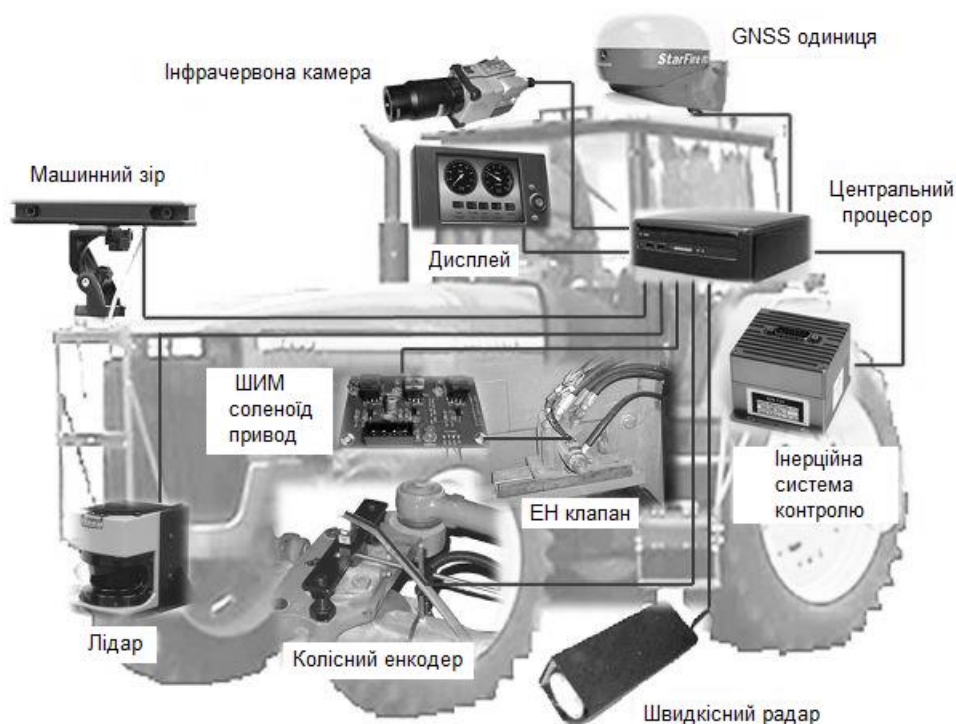


Рис. 1.10. Сенсорна архітектура інтелектуальних транспортних засобів

На рис. 1.11 зображено схему автомобільного крана, який обладнано мехатронними системами управління та контролю.

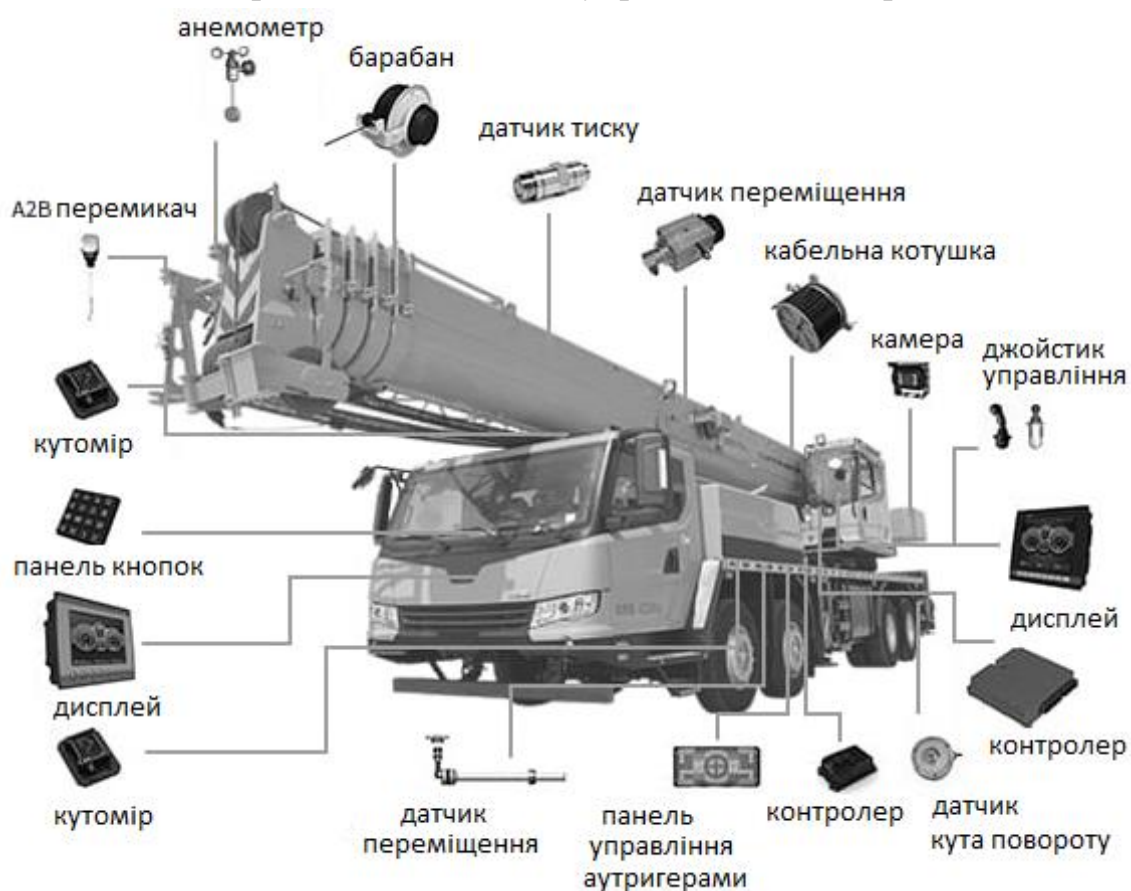


Рис. 1.11. Мехатронна система автомобільного крана

Зокрема, в такому автомобільному крані завдяки інтеграції електроніки реалізовано функції обмежувача вантажного моменту величиною 1600 тонн, автоматичного вирівнювання виносних опор, інтелектуального регулювання пневматичної підвіски, контролювання висоти підймання вантажу та швидкості повороту поворотної частини, а також систему інтерактивної діагностики. Комп'ютерна система такого крана дає змогу виконувати завантаження записів у реальному часі – 100 000 записів, а також контролювати режими роботи лебідки.

На рис. 1.12 зображено схему одного з варіантів архітектури мехатронної системи одноківшового екскаватора.

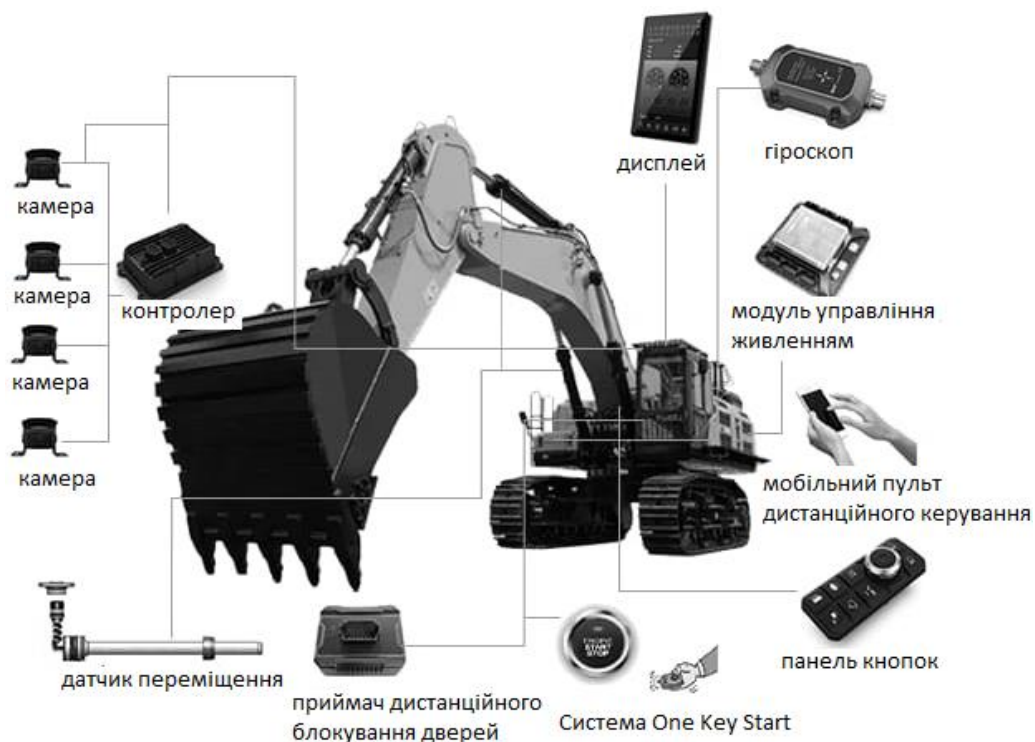


Рис. 1.12. Мехатронна система одноківшового екскаватора

Запропонована мехатронна система одноківшового екскаватора завдяки інтелектуальній системі гідроциліндра для визначення положення екскаватора дає змогу застосовувати екскаватор в обмеженій робочій зоні. Окрім того, комп'ютерна система керування обмежує крутний момент під час роботи екскаватора, управляє оптимізацією електроспоживання, відслідковує температурний режим машини та на основі локальної технології Wi-Fi і Bluetooth реалізовує дистанційний запуск двигуна машини, блокування дверей та налаштування роботи.

На рис. 1.13 зображено схему мехатронної системи машини для свердління тунелів. Мехатронна система такої машини забезпечує точне

свердління та автоматичне позиціонування, точне керування кутом рушійної балки та глибини отвору, невелике надрізання, може виконувати розширений геологічний аналіз, а саме визначати тиск, швидкість буріння та швидкість обертання бурильних труб, автоматично записувати та зберігати інформацію про буріння, що створює теоретичну основу для подальшої оптимізації.

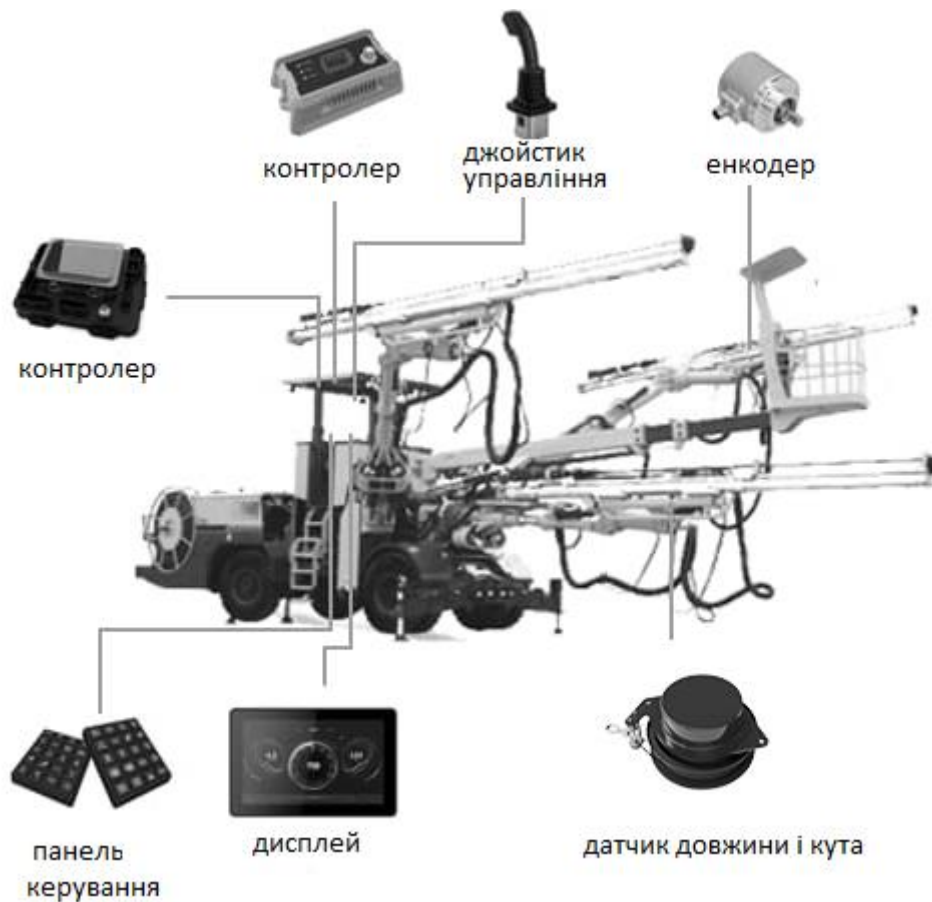


Рис. 1.13. Мехатронна система машини для прокладання тунелів

Складними мехатронними системами оснащують асфальтоукладачі, дорожні бетоноукладачі та бетоноформувальні машини. На рис. 1.14 зображено конструкцію асфальтоукладача з ковзною формою, який оснащено інтелектуальними системами розширення рами для зміни ширини дорожнього полотна, позиціонування опорних аутригерів та інтелектуальним рульовим управлінням із повністю кермовою гусеницею.

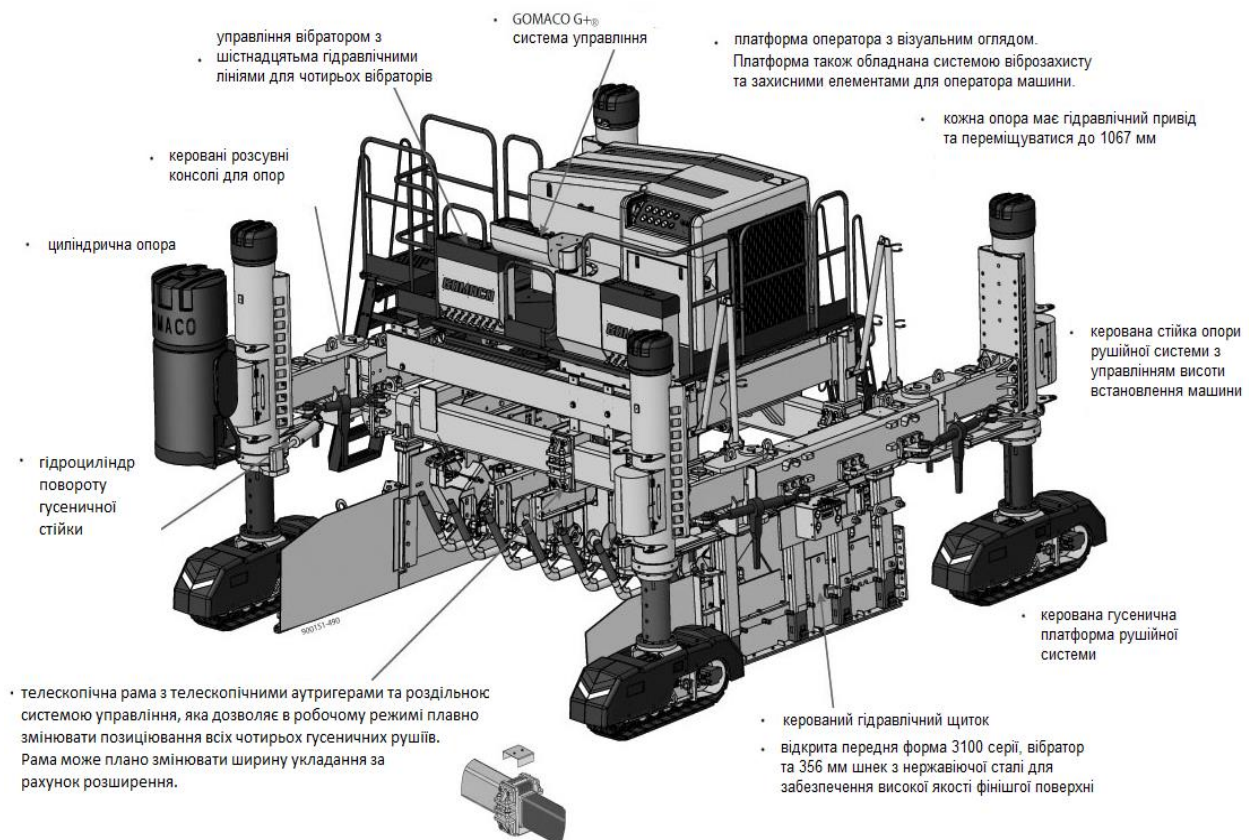


Рис. 1.14. Мехатронна система асфальтоукладача GOMACO GP3

Комп'ютерна система асфальтоукладача GOMACO GP3 з ковзною формою може автоматично визначати зміну ширини дорожнього полотна та контролювати товщину шару асфальтобетонної суміші, яку укладають. Інтелектуальні гідравлічні телескопічні елементи рами в передній і задній частинах головної рами забезпечують подвійну телескопічну рухомість. Інтелектуальні циліндри здатні контролювати товщину шару полотна і його нахил та передають інформацію в систему кермового керування й індивідуального контролю швидкості та радіуса повороту. Автоматичний розвал вбудовано в раму, оскільки вона подовжується за допомогою роликової рами. Інтелектуальне позиціонування опорних аутригерів включає поворотні приводи з датчиками обертання на шарнірних консолях кожного з чотирьох аутригерів асфальтоукладача. Поворотні приводи з датчиками обертання розміщено на кожній з чотирьох гусениць асфальтоукладача, що забезпечує найкращу технологію інтелектуального керування та дає змогу точно визначати розміщення та положення колії, розміри радіусів повороту та автоматично контролювати напрямок і швидкість руху.

Мехатронну систему комп'ютерної діагностики кар'єрного екскаватора зображено на рис. 1.15.

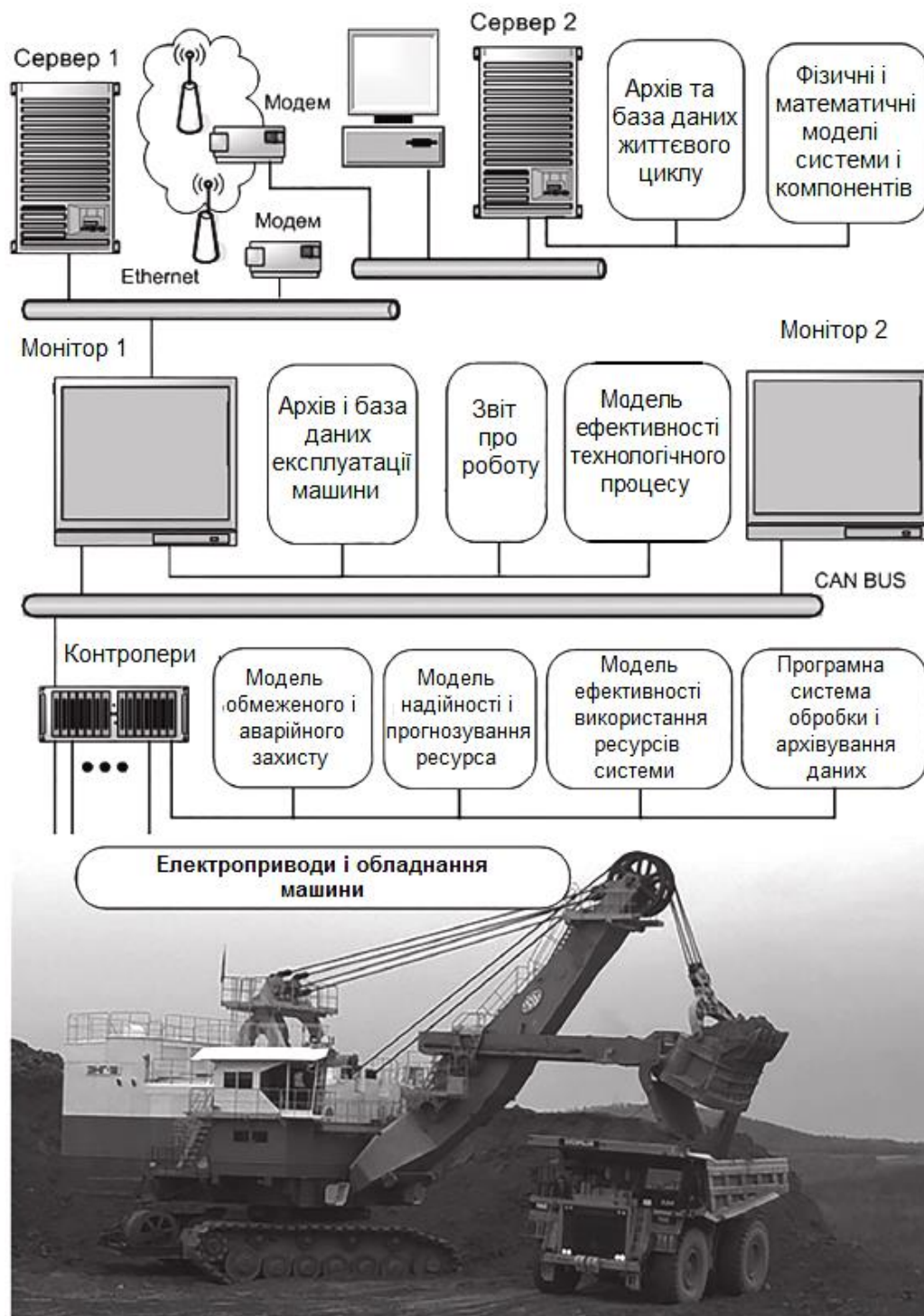


Рис. 1.15. Приклад мехатронної система комп'ютерної діагностики кар'єрного екскаватора

До складу машини з комп'ютерним керуванням належать такі основні компоненти:

- механічний пристрій, кінцевим ланцюгом якого є робочий орган;

- блок приводів, що має силові перетворювачі й виконавчі двигуни;
- пристрій комп'ютерного керування, верхнім рівнем для котрого є людина-оператор, або інший комп'ютер, що входить в комп'ютерну мережу;
- датчики, призначені для передавання в пристрій керування інформації про фактичний стан блоків машини й рух мехатронної системи.

Таким чином, наявність трьох обов'язкових частин: механічної (точніше, електромеханічної), електронної та комп'ютерної, які пов'язані енергетичними й інформаційними потоками, є первинною ознакою, яка вирізняє мехатронні системи. До електромеханічної частини належать механічні ланцюги та передачі, робочий орган, електродвигуни, сенсори й додаткові електротехнічні елементи (гальма, муфти). Механічний пристрій призначений для перетворення рухів ланцюгів у рух робочого органа.

Робоче середовище, в якому працює мехатронна система, є однією зі складових, яка визначатиме тип, конструкцію і правила застосування мехатронної системи. Середовища, в яких працюють механізми будівельних машин, можуть бути детерміновані та недетерміновані. До детермінованих належать середовища, для яких параметри збурювальних впливів і характеристики об'єктів робіт можуть бути заздалегідь визначені з достатньою для проєктування мехатронної системи величиною/ступенем точності. Деякі середовища є недетермінованими за своєю природою (наприклад, екстремальні середовища: підводні, підземні тощо). Характеристики середовищ, як правило, можуть бути визначені за допомогою аналітико-експериментальних досліджень і методів комп'ютерного моделювання. Наприклад, з метою оцінювання сил різання під час роботи землерийної машини виконують серії експериментів на спеціальних дослідних установках, параметри вібраційних впливів вимірюють на вібростендах з подальшим формуванням математичних і комп'ютерних моделей збурювальних впливів на підставі експериментальних даних. Проте для організації та виконання подібних досліджень зазвичай потрібні занадто складні вимірювальні технології й високовартісна апаратура. Тому в таких випадках доцільно застосовувати методи адаптивного керування, які дають змогу автоматично коригувати закон руху мехатронної системи безпосередньо під час виконання робіт.

Розвиток мехатроніки охоплює:

- постійне вдосконалення технологій в таких галузях, як механіка, електроніка та інформаційні технології й використання нових матеріалів (вуглецеві нанокомпозити, суперпровідники, п'єзоелектричні, магнітно-й фотоактивні, фоточутливі та самовідновлювальні матеріали), сенсори, приводи, а також програмне забезпечення для управління (MELSOFT MR Configurator 2, Simcenter Amesim) та обробки даних (Azure Microsoft, RStudio, Wolfram Alpha, Motorizer Mitsubishi Electric);
- створення інтелектуальних мехатронних модулів з інтегрованими електромеханічними пристроями, контролерами і датчиками;
- створення систем керування з відкритою архітектурою (типу PC-NC) на основі персональних комп'ютерів (ROS, Node-RED, MachineKit);
- мініатюризацію та компактність;
- впровадження штучного інтелекту в мехатронні системи сприяє їхній більшій адаптивності та здатності до автономного ухвалення рішень;
- розроблення розподілених систем керування рухом механізмів з кількома двигунами та об'єднання в мережі через стандартні інтерфейси і протоколи (PCI, CANbus, Profibus, USB, WiFi);
- впровадження енергоефективних технологій, використання відновлюваних джерел енергії та вдосконалення енергоощадних рішень;
- розвиток «Інтернету речей» (IoT), де мехатронні пристрої можуть взаємодіяти та обмінюватися даними через мережу.

Контрольні запитання

1. Які ознаки мехатронної системи?
2. Що таке мехатронна система?
3. Які складові мехатронної системи?
4. Що вирізняє мехатронну систему від автоматизованої системи?
5. Які пристрої належать до складу мехатронної системи бульдозера?
6. Які пристрої є в складі мехатронної системи екскаватора?
7. Які пристрої належать до мехатронної системи автомобільного крана?
8. Які пристрої належать до складу транспортної системи?
9. На які підсистеми поділяють мехатронні системи?
10. Які вимоги ставлять до мехатронних систем?
11. Які рівні обробки даних існують в мехатронних системах?
12. У чому полягає суть інформаційної інтеграції?

Лекція 2. Моделі мехатронних модулів будівельної техніки

Мехатронні модулі руху (ММР) є функціональними компонентами складних мехатронних систем. На рис. 2.1. наведено структурну схему ієрархії відомих мехатронних модулів, яка систематизує склад і ступінь інтеграції таких елементів.

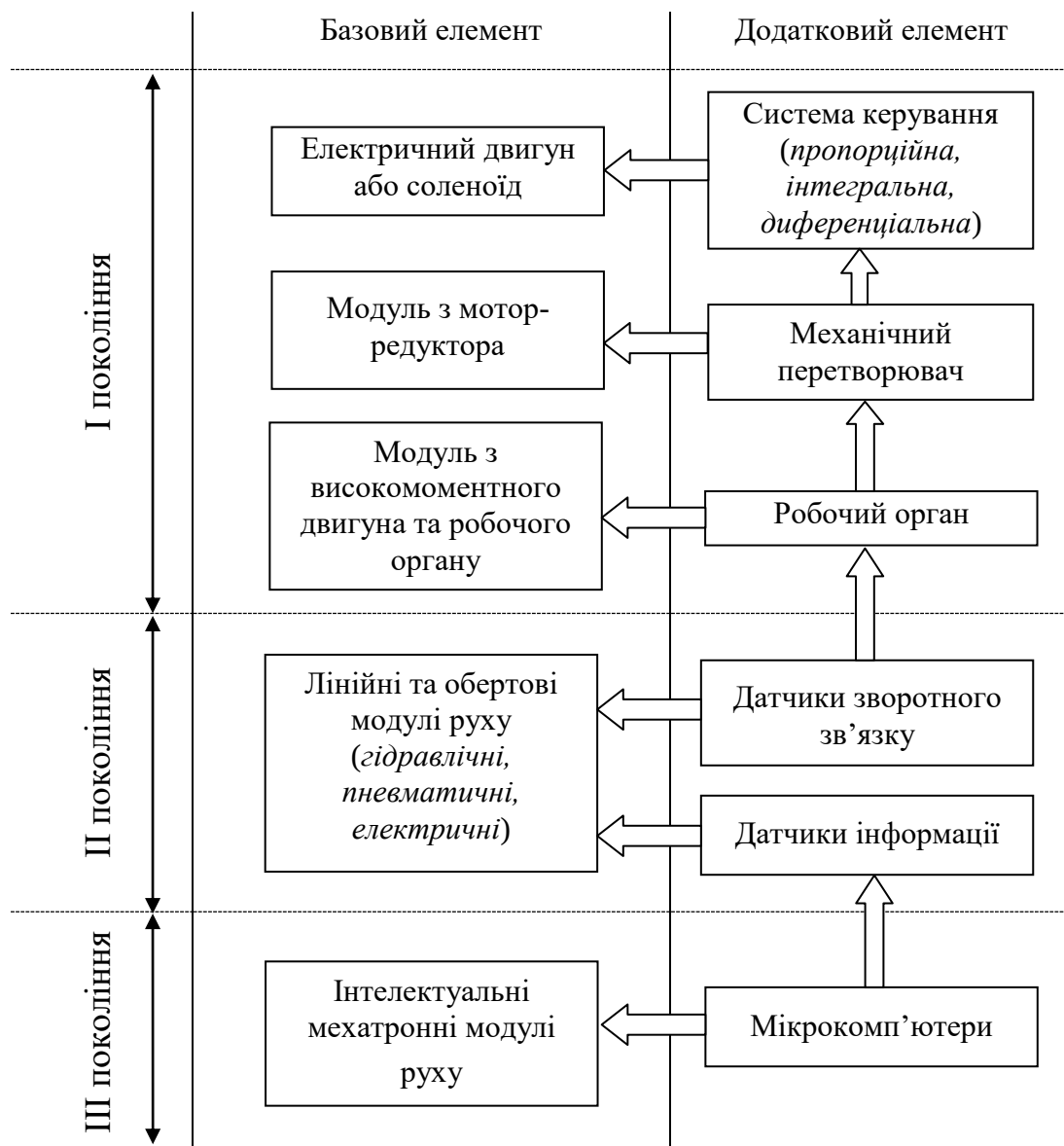


Рис. 2.1. Ієрархія поколінь мехатронних модулів

Електродвигуни та соленоїди є найпростішими елементами, на основі яких можна створити мехатронний модуль руху.

Електродвигуни класифікують таким чином (рис. 2.2).

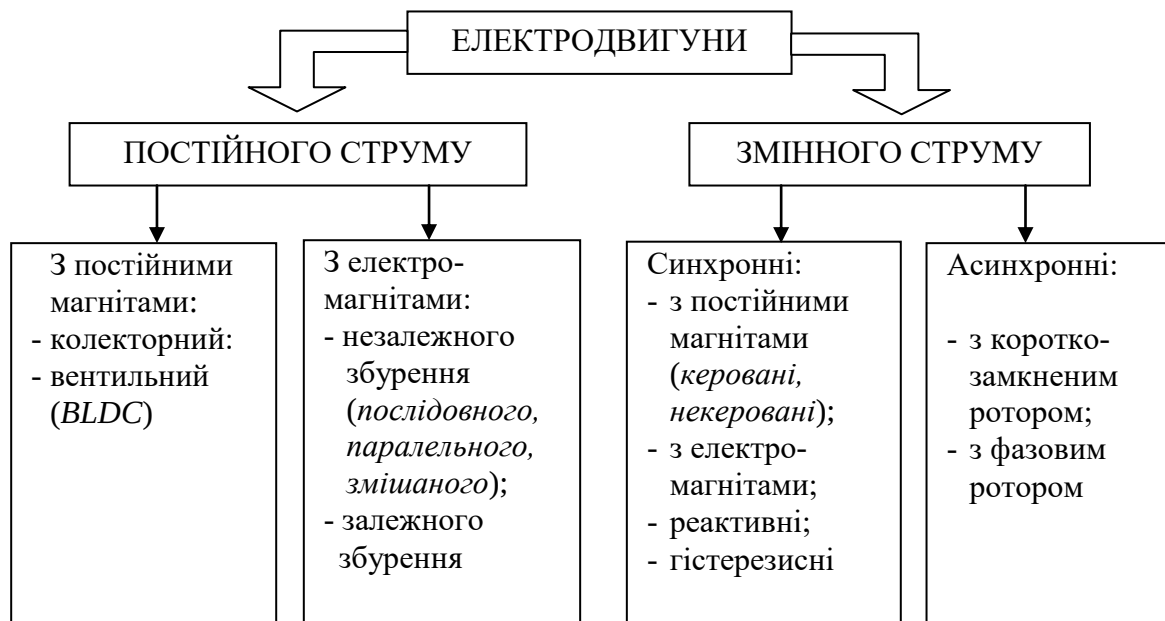


Рис. 2.2. Класифікація електродвигунів

Електродвигуни являють собою електродинамічні перетворювачі змінного або постійного магнітного поля в механічну роботу, зазвичай обертання ротора (класичний електродвигун). Існують також лінійні двигуни, де замість обертання ротора відтворюється його лінійне переміщення вздовж деякої напрямної. Пускова характеристика електродвигунів описується їхньою механічною характеристикою.

Обертальний момент, що утворюється електромагнітними силами, які діють на всі провідники обмотки якоря двигуна постійного струму, виражається формулою

$$M = \frac{D\gamma lBN}{2} \cdot i = k_F \Phi \cdot i, \quad (2.1)$$

де D – діаметр якоря електродвигуна; l – довжина котушки на виток; B – щільність магнітного потоку; N – кількість витків у котушці; i – величина сили струму; γ – коефіцієнт використання котушки; Φ – магнітний потік двигуна; k_F – коефіцієнт, що залежить від конструкції ротора двигуна.

Кутова швидкість обертання вала електродвигуна постійного струму в загальному може бути визначена за формулою

$$\omega = \frac{U - R \cdot i}{k\Phi}, \quad (2.2)$$

де Φ – магнітний потік, Вб; R – опір обмотки якоря, Ом; i – сила струму в обмотці якоря, А; k – конструктивний коефіцієнт.

В електродвигунах постійного струму характеристика крутного моменту на вихідному валу буде пропорційна силі струму на вході двигуна. В таких електродвигунах перетворення постійного струму на змінний зможе відбуватися через механічні щітки, розміщені на роторі, який обертається, або через електронну систему комунікації, тоді щітки для таких двигунів не потрібні.

Якщо електродвигун обладнати датчиком, тоді можна отримати найпростіший мехатронний модуль руху (рис. 2.3).

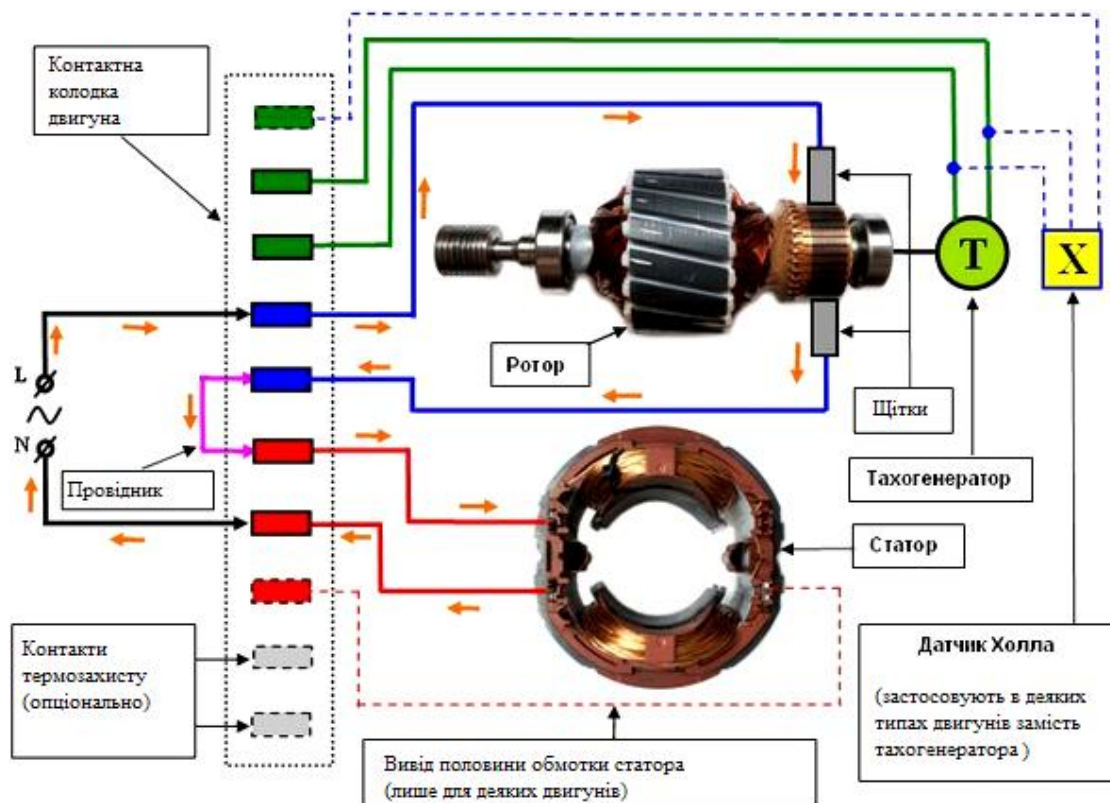


Рис. 2.3. Принципова схема мехатронного модуля руху на основі колекторного електродвигуна

Обертальний електромагнітний момент на валу асинхронного двигуна змінного струму виражається такою формулою

$$M \approx \frac{4,44m_2 f_1 w_2 k_2}{\omega_1} \cdot \Phi \cdot i_2 \cos \psi_2, \quad (2.3)$$

де ω_1 – кутова частота струму в обмотці статора, рад/с; m_2 – кількість фаз обмотки ротора (для трифазного двигуна $m_2 = 3$); i_2 – сила струму в обмотці ротора, А; ψ_2 – кут зсуву фаз між ЕРС та струмом ротора; f_1 – циклічна частота струму в обмотці статора, Гц; w_2 – кількість послідовно з'єднаних витків обмотки ротора; k_2 – обмотувальний коефіцієнт ротора; Φ – магнітний потік статора (амплітудне значення), Вб.

Найбільш поширеним асинхронним електродвигуном є двигун з короткозамкненим ротором. Під час розгону асинхронного двигуна з короткозамкненим ротором змінюється опір обмотки ротора, а отже, залежно від частоти обертання (від величини ковзання) змінюється й обертальний момент. Перемикання кількості полюсів – це найдешевший спосіб зміни частоти обертання приводу з асинхронним двигуном, тому такі двигуни використовують у приводних системах підйомно-транспортного обладнання, при цьому висока частота обертання потрібна для швидкого ходу, а низька – для позиціонування.

Після перемикання обмотки статора з 2-полюсного на 8-полюсне ввімкнення, частота обертання не відразу знижується до синхронної, і двигун короткочасно працює в генераторному режимі. Завдяки перетворенню кінетичної енергії на електричну відбувається короткочасне гальмування від високої частоти обертання до низької, яке виконується без втрати потужності та спрацювання деталей. За такого гальмування досягається крутий момент, величину якого в першому наближенні розраховують за формулою:

$$M \approx (2...2,5)M_{A1}, \quad (2.4)$$

де M_{A1} – пусковий момент двигуна.

Номінальна потужність електродвигуна, вказана в його технічних характеристиках, залежить від температури навколишнього середовища та висоти над рівнем моря (рис. 2.4). Номінальна потужність дійсна за температури навколишнього середовища до 40 °С та висоти не більш ніж 1000 м над рівнем моря. У разі відхилення від цих даних номінальну потужність слід знизити, перерахувавши її за формулою

$$N_{n-1} = N_n f_T f_{H1}, \quad (2.5)$$

де N_n – номінальна потужність, кВт; f_T – температурний коефіцієнт перерахунку; f_H – висотний коефіцієнт перерахунку.

Номінальна частота обертання двигуна n_N за номінальної потужності в рушійному режимі завжди менша за синхронну частоту обертання n_s унаслідок ефекту ковзання, тобто різниці синхронної та фактичної частот обертання. Величину ковзання визначають за формулою, %

$$S = \frac{n_s - n_N}{n_s} \cdot 100\%, \quad (2.6)$$

де n_N – номінальна частота обертання; n_N – синхронна частота обертання.

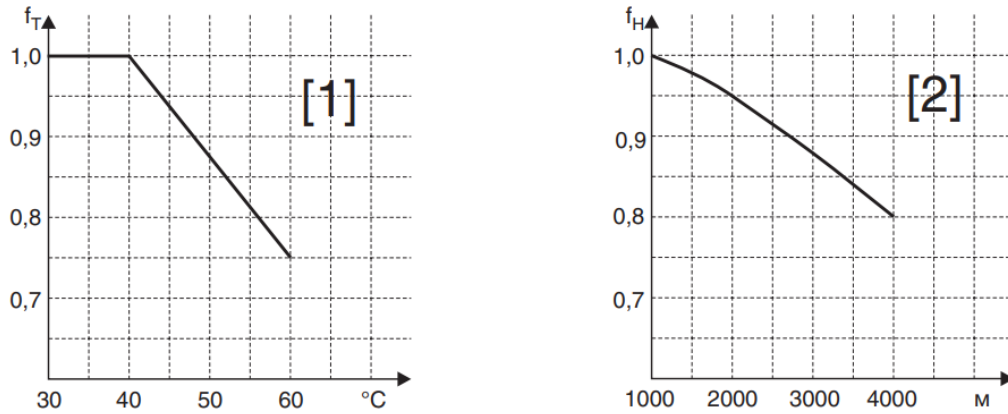


Рис. 2.4. Зниження потужності електродвигуна залежно від температури навколишнього середовища (1) та висоти над рівнем моря (2)

Безщітковий електродвигун постійного струму (BLDC), або електронно-комутований двигун, – це синхронний електродвигун постійного струму, в якому використано електронний контролер для перемикання постійного струму на обмотки двигуна, створюючи магнітні поля, які обертаються в просторі, що призводить до руху ротора з постійним магнітом. Конструкції системи безщіткового двигуна зазвичай подібні до синхронного двигуна з постійними магнітами (PMSM), але також можуть бути реактивними або асинхронними. Існують конструкції двигунів зі статором всередині ротора, з ротором, який обертається всередині статора, або аксіальні, у яких ротор і статор плоскі та паралельні (рис. 2.5).

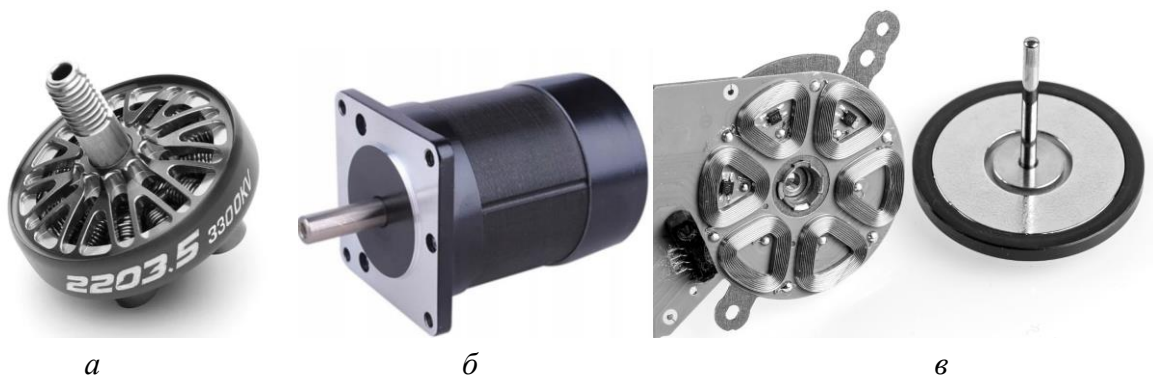


Рис. 2.5. Типи вентильних електродвигунів: *а* – зі статором всередині ротора; *б* – зі статором зовні ротора; *в* – аксіальний

В конструкціях BLDC двигунів можуть бути використані неодимові магніти. Контролер регулює фазу та амплітуду імпульсів постійного струму для керування швидкістю та крутним моментом двигуна. У безщіткових двигунах постійного струму електронна система, яка замінює механічні контакти комутатора, складається з електронного датчика, який визначає кут повороту ротора та керує напівпровідниковими перемикачами, такими як транзистори, які перемикають струм через обмотки або змінюють напрямок струму. У деяких двигунах електронна система може вимикати струм за заданих кутів повороту, щоб електромагніти створювали крутний момент в одному напрямку.

Комутація безщіткового двигуна може бути реалізована програмним забезпеченням за допомогою мікроконтролера або альтернативно, за допомогою аналогових або цифрових схем. На рис. 2.6 наведено схему комутації BLDC двигуна з електронним контролером та показано діаграму програмного ввімкнення контактів такої схеми.

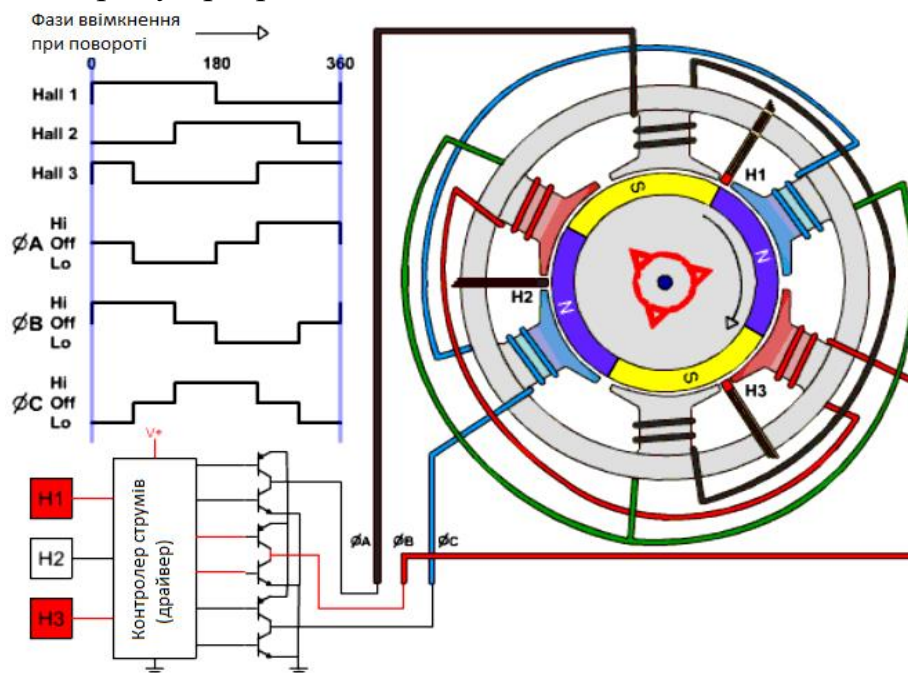


Рис. 2.6. Схема комутації BLDC двигуна з контролером і транзисторні ключі

Є дві загальні конфігурації електричних обмоток. Конфігурація «трикутник», яка з'єднує три обмотки в одну схему, а живлення подається на кожне зі з'єднань. Конфігурація «зірка» з'єднує всі обмотки в центральній точці, а живлення подається на решту кінців кожної з обмоток. Двигун з обмотками в трикутній конфігурації має низький крутний момент на низькій швидкості, але може забезпечити вищу максимальну швидкість. Конфігурація «зірка» забезпечує високий

крутний момент на низькій швидкості, але не таку високу максимальну швидкість. Обмотка «зіркою» зазвичай більш ефективна. Обмотки, з'єднані «трикутником», не перешкоджають високочастотним паразитним електричним струмам повністю циркулювати всередині двигуна. Обмотка, з'єднана «зіркою», не містить замкнутого контуру, в якому можуть протікати паразитні струми, що запобігає втратам. Крім вищого опору конфігурації «зірки», з погляду контролера, дві конфігурації обмоток можна розглядати абсолютно однаково.

Кроковий електродвигун (рис. 2.7) – це різновид безщіткового електродвигуна, в якому імпульсне живлення електричним струмом призводить до того, що ротор двигуна не обертається неперервно, а виконує поворот на заданий кут. Кут повороту ротора залежить від кількості поданих імпульсів струму, а кутова швидкість ротора точно дорівнює частоті імпульсів, помноженій на кут повороту ротора за один цикл роботи двигуна.

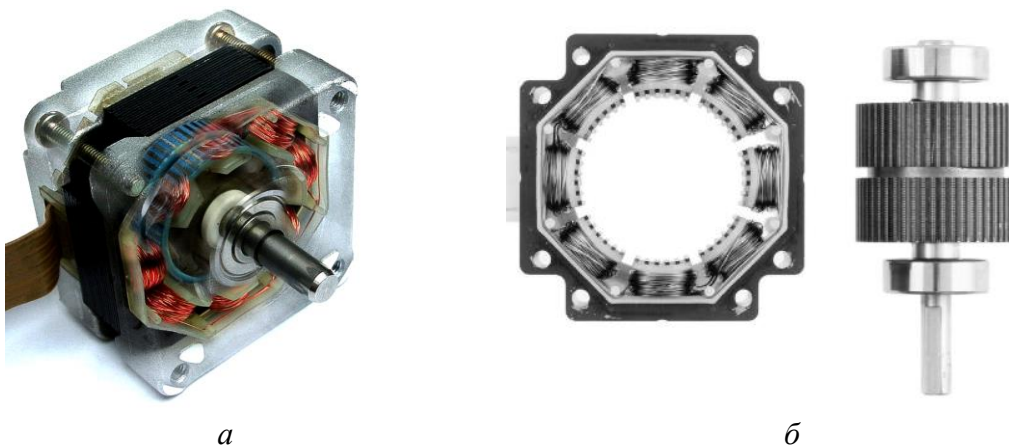


Рис. 2.7. Загальний вигляд (а) та вигляд статора і ротора (б) крокового електродвигуна

За конструкцією ротора виділяють три типи крокових двигунів: реактивний, з постійними магнітами, гібридний.

Реактивний кроковий двигун (рис 2.8) – це синхронний двигун, статор якого зазвичай має шість полюсів і три фази (по два полюси на фазу), а ротор – чотири полюси. За такої конструкції двигуна крок дорівнює 30° . На відміну від інших крокових двигунів вимкнений реактивний кроковий двигун не має фіксувального моменту, що гальмує ротор під час його обертання.

Кроковий двигун із постійними магнітами має ротор з постійними магнітами (активний ротор) та статор з двома фазами (рис. 2.9).

Порівняно з реактивними крокові двигуни з активним ротором створюють великі обертові моменти, потрібні для фіксації ротора при знятті керівного сигналу. Недолік двигунів із постійними магнітами – великий кутовий крок ($7,5 \dots 90^\circ$). Кроковий двигун із постійними магнітами може мати різну схему з'єднання обмоток статора.

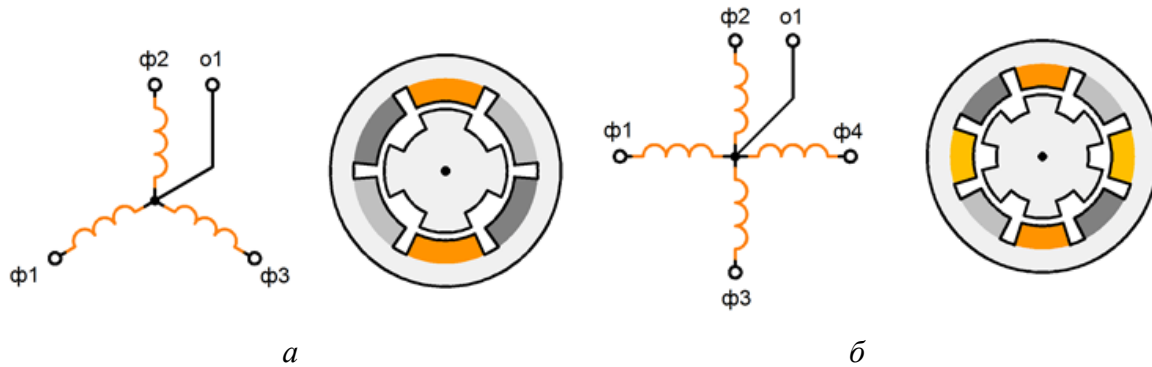


Рис. 2.8. Схема обмоток трифазного з кроком 30° (а) та чотирифазного з кроком 15° (б) реактивних крокових двигунів

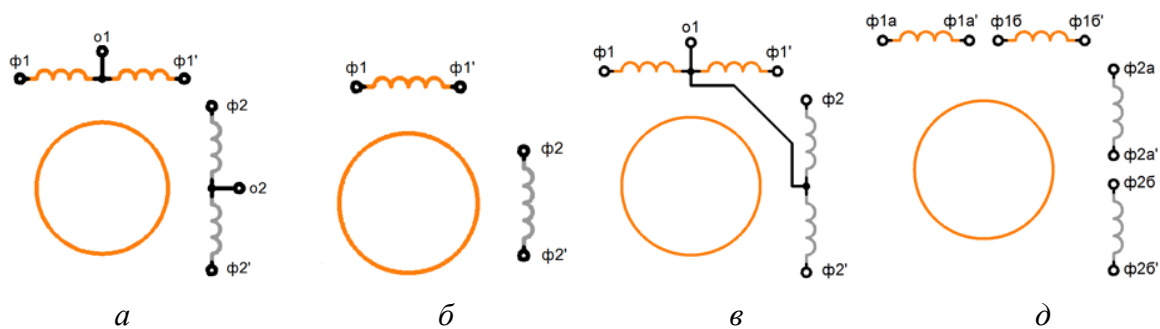


Рис. 2.9. Схеми 6-контактного уніполярного (а), 2-контактного біполярного (б), 5-контактного двофазного (в) та 8-контактного крокового двигуна (д)

Обмотки крокових двигунів можуть мати відгалуження в центрі для роботи з однополярною схемою керування. Двополярне керування потрібне для живлення обмоток без центрального відгалуження. Двополярні двигуни також мають одну обмотку на фазу. Для того щоб змінити магнітну полярність полюсів, треба змінити напрям струму в обмотці, а для цього ускладнюють схему керування, зазвичай з Н-мостом. Біполярні двигуни мають краще співвідношення потужність/об'єм, ніж уніполярні.

Кроковим двигуном з чотирма виводами може керувати лише біполярним способом, 6-контактний двигун призначений для керування уніполярним способом, незважаючи на те, що він також може бути керований біполярним способом, якщо ігнорувати центральні виходи, а 5-контактний двигун може керуватися тільки уніполярним способом,

оскільки загальний центральний вихід з'єднує обидві фази. Конфігурація двигуна з вісьмома контактами трапляється рідко, але забезпечує максимальну гнучкість. Такий двигун може бути під'єднаний для керування також як 6- або 5-контактний. Пара обмоток може бути під'єднана послідовно для високовольтного біполярного керування малими струмами або паралельно для низьковольтного керування великими струмами.

Гібридний кроковий двигун (рис. 2.10) створено з метою поєднати найкращі властивості обох крокових двигунів – реактивного і з постійними магнітами, що дало змогу досягти меншого кута кроку. Ротор гібридного крокового двигуна являє собою циліндричний постійний магніт, намагнічений уздовж поздовжньої осі з радіальними зубами з магнітом'якого матеріалу. Статор зазвичай має дві або чотири фази, розподілені між парами явно виражених полюсів. Обмотки статора можуть мати центральне відгалуження для уніполярного керування. На роторі розміщують 48 зубів на одній секції, які зміщують на половину зубцевого поділу λ щодо іншої секції. Завдяки зміщенню ротор фактично має 96 полюсів. Така особливість конструкції дає змогу керувати з 48, 96 та 192 кроками.

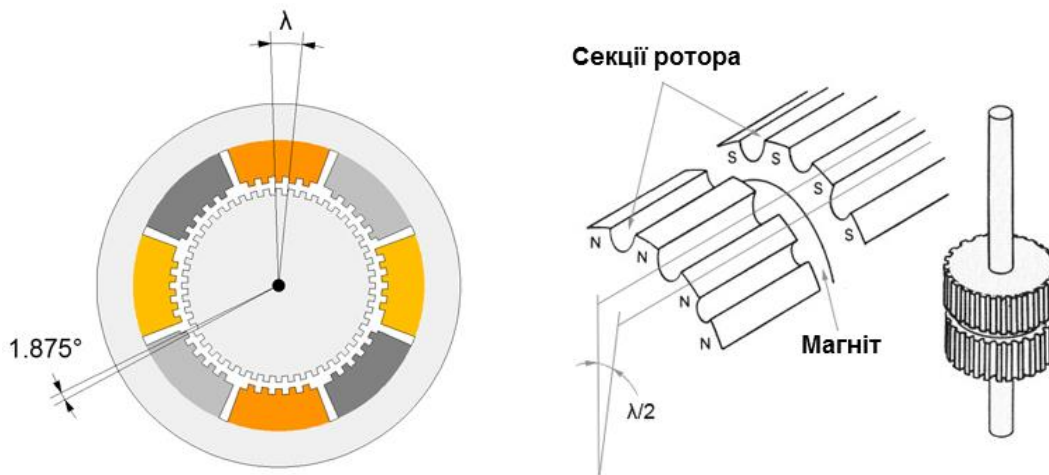


Рис. 2.10. Гібридний кроковий двигун

Соленоїд – це електромагнітний перетворювач лінійної дії, що складається з котушки дроту, яку намотано на циліндричну поверхню. Принцип роботи всіх електромагнітних перетворювачів (тягових або обертових) ґрунтується на силі магнітного поля, що діє на тіло, яке завдяки своїм властивостям може змінювати магнітне поле та створювати магнітну силу.

Соленоїди (рис. 2.11) застосовують електромагніти як поступальний силовий електропривід. Якщо довжина соленоїда більша за його діаметр, тоді внаслідок протікання струму всередині котушки виникає однорідне магнітне поле, спрямоване вздовж її осі. Матеріали, які взаємодіють з таким магнітним полем та мають достатню магнітну проникність, будуть намагнічуватися. Магнітна проникність матеріалу $\mu = \mu_r \mu_0$ складається з добутку магнітної проникності вакууму (абсолютної проникності) $\mu_0 = 4\pi 10^{-7}$ Гн/м та відносної магнітної проникності μ_r .

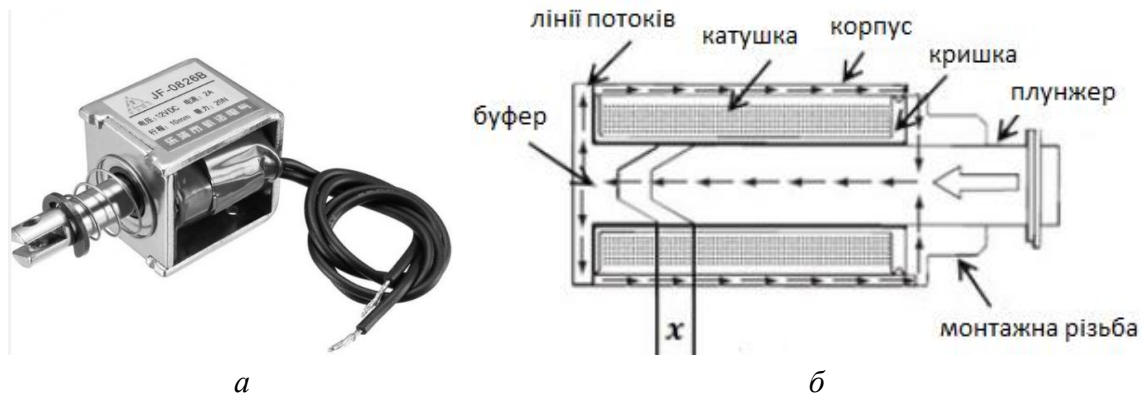


Рис. 2.11. Загальний вигляд (а) та конструктивна схема (б) лінійного соленоїда

Індуктивність соленоїда:

$$L = \mu_0 n^2 V = \frac{\mu_0 z^2}{4\pi l}, \quad (2.7)$$

де μ_0 – магнітна проникність вакууму; $n = N/l$ – кількість витків на одиницю довжини соленоїда; $z = \pi d N$ – довжина провідника намотаного на соленоїд; $V = Sl$ – об’єм соленоїда; $S = \pi d^2/4$ – площа поперечного перерізу соленоїда; l – довжина соленоїда.

Якщо прямий провідник зі струмом зігнути у гвинтову котушку (соленоїд) з N витків, тоді такий пристрій буде індукувати магнітне поле. Якщо довжина котушки l набагато більша за її діаметр D , тоді щільність потоку відповідає правилу правої руки, а величина магнітного поля всередині котушки приблизно дорівнюватиме:

$$B = \mu \frac{N}{l} i. \quad (2.8)$$

Більшість лінійних соленоїдів складається з нерухомої залізної рами (статора), котушки (соленоїда) і феромагнітного плунжера (якоря) в центрі котушки. Плунжер рухається, щоб збільшити зчеплення потоку

завдяки зменшенню повітряного зазору між плунжером і нерухомою рамою. Магнітна генерована сила приблизно пропорційна квадрату величини прикладеної сили струму i та обернено пропорційна квадрату повітряного зазору δ , який є ходом соленоїда, тобто:

$$F \propto \frac{i^2}{\delta^2}. \quad (2.8)$$

Обертові соленоїди використовують шарикопідшипники, які рухаються по похилих доріжках кочення для перетворення лінійного руху в обертовий (рис. 2.11). Коли котушка залишається під напругою, плунжерний вузол переміщується до статора та обертається, що визначається профілем доріжки кочення.



Рис. 2.11. Загальний вигляд поворотного соленоїда

Одним з основних елементів мехатронних систем є модулі руху на основі мотор-редуктора. У такому модулі руху поєднано приводний електричний двигун та індустріальний механічний редуктор (рис. 2.12).



Рис. 2.12. Мотор-редуктор: *а* – циліндричний; *б* – циліндричний з варіатором; 1 – електричний двигун; 2 – редуктор; 3 – вихідний вал редуктора

Для того щоб мотор-редуктор перетворити на мехатронний модуль, його потрібно обладнати датчиком та контролером. Якщо в таку мехатронну систему додати зворотний зв'язок контролю положення вихідного вала редуктора, отримаємо *сервосистему*, або *сервопривод*.

Сервопривод – це керований привод із системою управління, в якій згідно з відповідними змінами цілого значення відбувається зміна положення, орієнтації та інших параметрів керованого об'єкта.

Електричний сервопривід містить електродвигун з редукторним приводом, який може керовано змінювати кут повороту вихідного вала на задане значення й утримувати це положення (рис. 2.13). У більшості механізмів використано гідравлічні або пневматичні сервосистеми, однак останнім часом широкого застосовування набули електричні системи, завдяки зручності їхнього обслуговування. Для досягнення високої точності керування найчастіше використовують електродвигуни змінного струму з датчиками кута повороту, швидкості та напрямку повороту. Як блок управління використовують сервопідсилювач зі зворотним зв'язком.

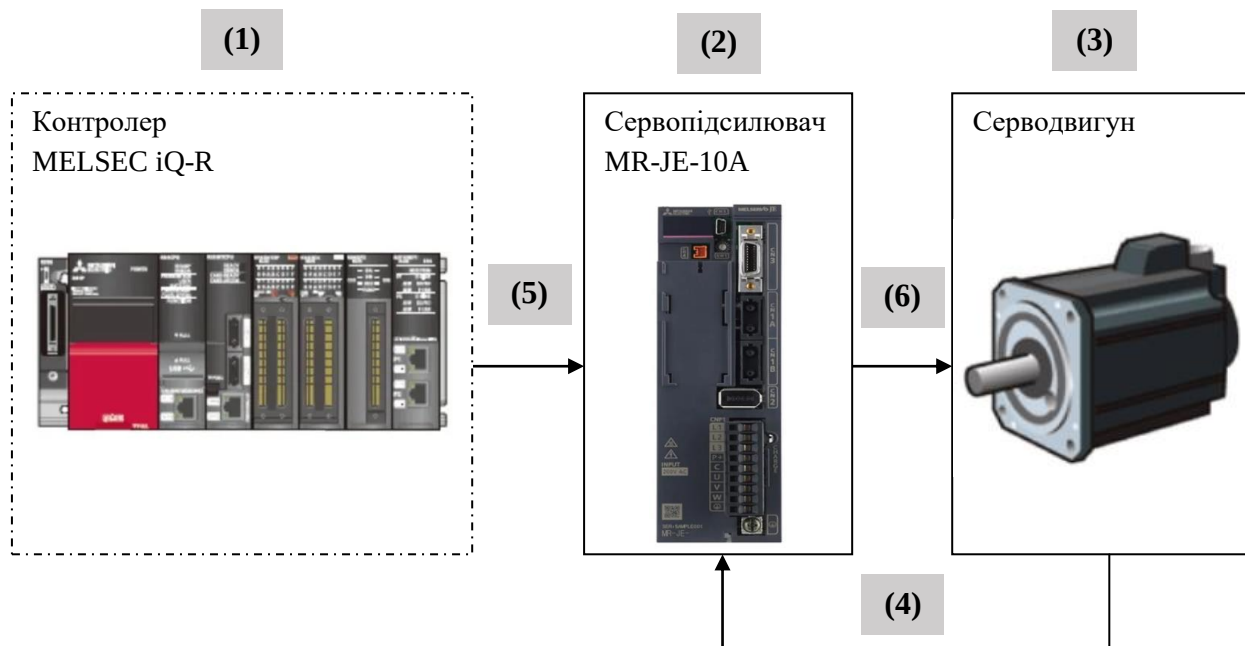


Рис. 2.13. Схема сервоприводу: 1 – командний блок; 2 – блок керування; 3 – блок приводу і датчика; 4 – зворотний сигнал; 5 – сигнали керування; 6 – електроживлення

Системи з блока приводу, датчика та сервопідсилювача можуть мати монолітну будову (рис. 2.14).



Рис. 2.14. Електросервоприводи монолітної будови з блока сервопідсилювача, привода та датчика

Під час роботи сервопривод отримує на вхід значення керівного параметра, блок керування порівнює це значення зі значенням на датчику і на основі порівняння виконує певну дію, наприклад: поворот, прискорення або сповільнення так, щоб значення з внутрішнього датчика стало якомога ближчим до значення зовнішнього керівного параметра. Керівним сигналом для електросервоприводу є імпульси постійної частоти і змінної ширини. Залежно від довжини імпульсів визначається положення вихідного вала сервоприводу. Електронна система сервоприводу, яка отримує імпульси на вхід через власний генератор імпульсів, виробляє електричний імпульс, тривалість якого визначається через потенціометр (внутрішній датчик сервоприводу). Інша частина схеми порівнює тривалість двох імпульсів, і якщо тривалість різна, вмикається електродвигун. Напрямок обертання визначається довжиною коротшого з двох порівнюваних імпульсів. Якщо довжина імпульсів однакова, електродвигун зупиняється.

Найчастіше в аматорських сервоприводах імпульси виробляються з частотою 50 Гц. Це означає, що імпульс генерується та реєструється один раз на 20 мс. За тривалості імпульсу 1520 мкс вихідна ланка сервоприводу повинна зайняти середнє положення (рис. 2.15). Збільшення або зменшення довжини імпульсу змусить сервопривід повернутися за годинниковою або проти годинникової стрілки відповідно. При цьому є верхня і нижня межі тривалості імпульсу.

Оптимальним способом плавного керування частотою обертання двигунів і мотор-редукторів змінного струму є використання перетворювачів частоти. Такі перетворювачі дають регульовану за частотою напругу, величина якої змінюється пропорційно до зміни

частоти (рис. 2.16). Перетворювачі частоти застосовують переважно для асинхронних електродвигунів.

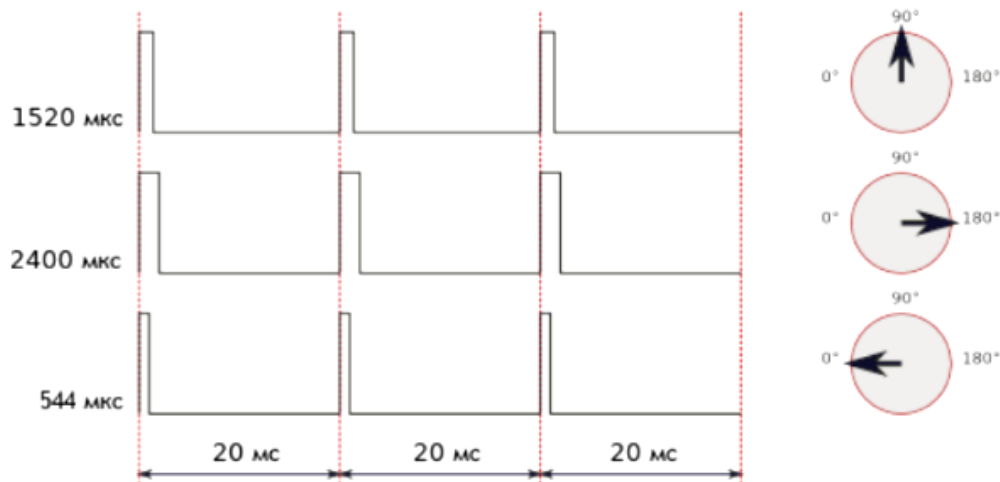


Рис. 2.15. Залежність кута повороту сервоприводу від тривалості керівних імпульсів



Рис. 2.16. Частотні перетворювачі

Частотний перетворювач змінює частоту живлення електродвигуна, яка приблизно може бути розрахована за формулою

$$n_{дв} = \frac{120 \cdot f}{w} (1 - S), \quad (2.9)$$

де f – частота джерела живлення, Гц; w – кількість пар полюсів електродвигуна (залежить від конфігурації двигуна, вказаної в маркуванні, наприклад, 4Р використовують для 4-полюсних двигунів); S – коефіцієнт ковзання (за номінальної роботи асинхронних двигунів зазвичай становить від 0,03 до 0,05, а в разі зупиненого двигуна $S = 1$).

Залежно від потрібної навантажувальної характеристики (рис. 2.17) обирають характеристику керування частотного перетворювача.

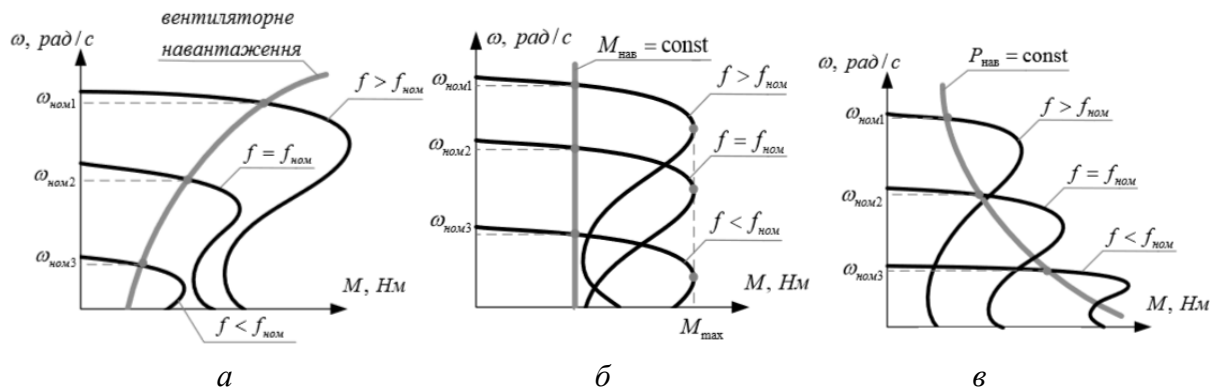


Рис. 2.17. Види механічних характеристик: *а* – навантаження з крутним моментом, який збільшується прямо пропорційно квадрату частоти обертання ($U = f^2 B^*$); *б* – навантаження з постійним крутним моментом ($U = f B^*$); *в* – навантаження з крутним моментом, який обернено пропорційний частоті обертання ($U = \sqrt{f} B^*$); B – постійний коефіцієнт

Розрізняють два основні принципи керування перетворювачами частоти – *скалярний* і *векторний*. За скалярного керування відбувається формування фазних напруг на основі заданих значень амплітуди і частоти, які отримують шляхом широтно-імпульсної модуляції (ШІМ). Скалярне керування забезпечує постійну перевантажувальну здатність електропривода незалежно від частоти напруги, проте відбувається зниження моменту, що створюється двигуном за низьких частот (якщо $f < 0,1f_n$). Максимальний діапазон регулювання швидкості обертання ротора за незмінного моменту опору для електродвигуна зі скалярним керуванням сягає 1:10. Цей принцип є найбільш простим способом частотного керування, застосовують його для приводу механізмів, до яких не ставлять високих вимог щодо якості регулювання швидкості (рис. 2.17, *а*). Насамперед це стосується електроприводів насосів, вентиляторів, компресорів. Цей клас механізмів має широку потенційну здатність до енерго- і ресурсозбереження, що успішно реалізується завдяки впровадженню цього типу перетворювачів.

Недоліками скалярного керування без зворотного зв'язку (за закону керування $U/f = \text{const}$) є таке:

- низька якість регулювання швидкості як керівної, так і збудовувальної дії;

- неможливість режиму роботи на упор, тобто в моментах надмірних навантажень привод вимикається під дією струмового захисту;
- важко досягти керування потокозчепленням у функції моменту двигуна.

Скалярне керування може відбуватися шляхом регулювання струму статора, що порівняно з частотним скалярним керуванням напруги статора дає обмеження моменту двигуна на заданому рівні незалежно від швидкості двигуна.

Недоліки скалярного частотно-струмового керування:

- втрата контролю за моментом двигуна в перехідному режимі через велику інерційність контуру регулювання;
- незмінність магнітного потоку для статичного режиму роботи унаслідок формування струму статора у функції абсолютного ковзання;
- усереднення відображення системи трифазних струмів у зв'язку з контролем струму статора постійним випрямленим струмом.

Для дотримання складних законів керування використовують замкнені системи регулювання зі зворотним зв'язком.

Друга принципова схема керування – це система векторного керування, яка забезпечує характеристики асинхронного двигуна, близькі до регульованого електроприводу постійного струму. Цих властивостей системи досягають завдяки розділенню каналів регулювання потокозчеплення і швидкості обертання електродвигуна, що неможливо в разі використання скалярного керування. Відомо два основні класи систем векторного керування – без датчика швидкості на валу двигуна і системи зі зворотним зв'язком за швидкістю. За невеликих діапазонів зміни швидкості (не більш ніж 1:100) і вимог до точності, що не перевищують $\pm 0,5\%$, використовують векторне керування без датчика. Якщо швидкість обертання вала змінюється в широких межах (до 1:10000 і вище) і за вимог до високої точності підтримання швидкості обертання (до $\pm 0,02\%$ за частот обертання, менших за 1 Гц) або якщо є потреба в позиціонуванні вала, а також за потреби в регулюванні моменту на валу двигуна на дуже низьких частотах обертання, застосовують методи векторного керування зі зворотним зв'язком за швидкістю.

Типову схему під'єднання двигуна до частотного перетворювача наведено на рис. 2.18.

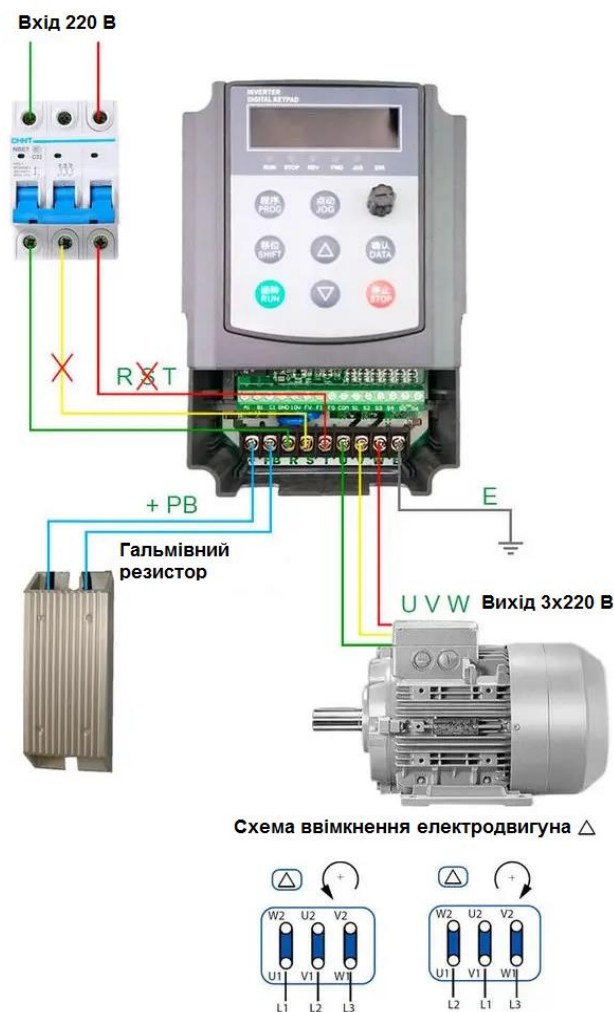


Рис. 2.18. Схема під'єднання електродвигуна до частотного перетворювача

Варто пам'ятати, що за зменшення частоти струму змінюється баланс активного та реактивного опору обмоток електродвигуна, а отже, зменшується крутний момент.

Існують електродвигуни із вбудованим частотним перетворювачем потужністю від 0,55 до 7,5 кВт (рис. 2.19). Основними перевагами таких двигунів є компактність та непотрібність кабелів великої довжини, а завдяки наявності в їхньому складі обчислювальних пристроїв вони здатні автономно виконувати переміщення робочих органів машин без постійного контролю з боку системи автоматизації верхнього рівня.

Лінійні, або електричні, циліндри – це електричні регульовані пристрої, які можуть переміщати шток по прямій лінії. Електроциліндри є альтернативою пневматичним або гідравлічним циліндрам. В лінійних електричних циліндрах обертальний рух вала електродвигуна перетворюється в лінійне переміщення за допомогою кулько-гвинтової передачі (рис. 2.20). Стандартне зусилля таких циліндрів становить до 40 кН, а спеціальних – до 178 кН.



Рис. 2.19. Двигуни та мотор-редуктори з вбудованими перетворювачами частоти

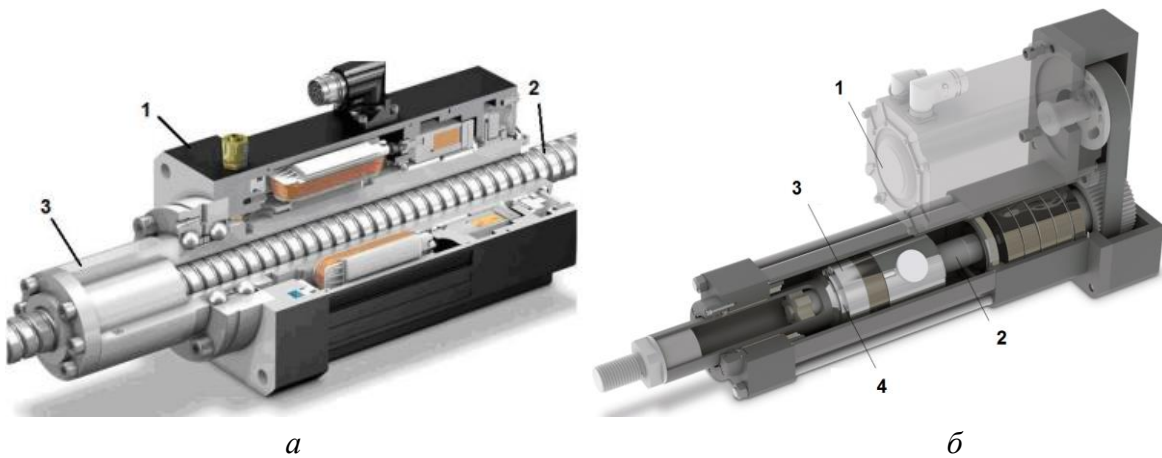


Рис. 2.20. Лінійні електроприводи: *а* – з вбудованим двигуном; *б* – з навісним двигуном; 1 – електродвигун; 2 – гвинт; 3 – гайка; 4 – кулькова гайка-упор

Гідравлічні та пневматичні сервосистеми – це керовані пристрої для управління механічною вихідною потужністю, які виконують регулювання кінематичних (керування положенням або швидкістю) або силових (керування силою, крутним моментом або тиском) величин. В гідравлічних системах регулювання вихідної величини досягається шляхом керування потужністю потоку рідини, а в пневматичних – стиснутого повітря. Робочий тиск у гідравлічних сервосистемах зазвичай становить від 150 до 300 бар (15-30 МПа), а у пневматичних систем значення тиску зазвичай нижче за 10 бар (1 МПа).

Системи гідравлічних сервоциліндрів складаються з таких вузлів: сервоциліндра, сервоблока керування, електроніки для регулювання та налаштування, датчика та допоміжних комунікаційних пристроїв (рис. 2.21).



Рис. 2.21. Будова гідравлічного сервоциліндра

Залежно від умов застосування виокремлюють такі типи гідравлічних сервоциліндрів:

- сервоциліндри з гідростатичним клиновим зазором опори штока;
- сервоциліндри з гідростатичними «кишенями» опори штока.

Сервоциліндри з гідростатичним клиновим зазором для опори штока (рис. 2.22) застосовують для швидкостей до 2 м/с за невеликих бічних навантажень (наприклад, від власної маси або сили інерції). Такі циліндри використовують для робочого тиску до 21 МПа і номінального зусилля від 1 до 4000 кН.

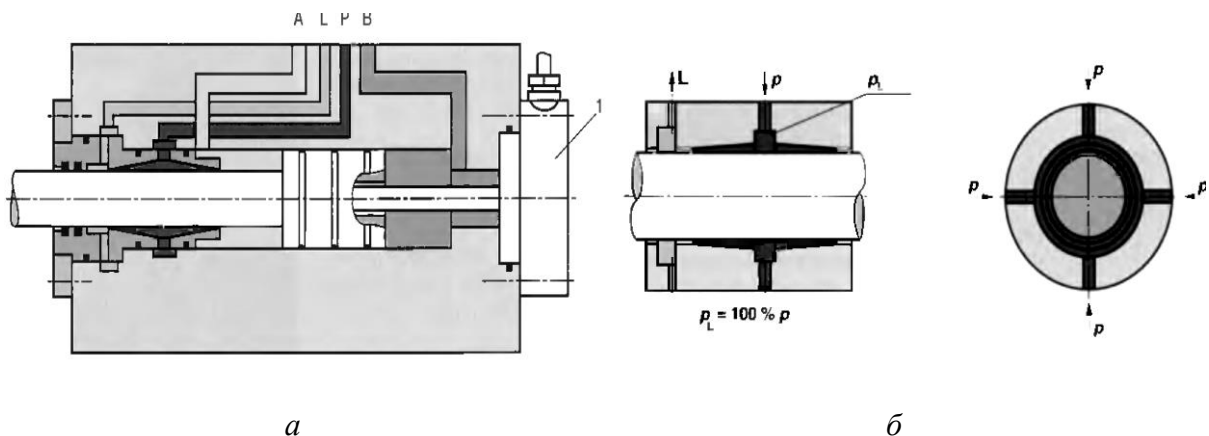


Рис. 2.22. Схеми сервоциліндра з опорою штока у вигляді гідростатичного клинового зазору (а) та опори штока (б)

Сервоциліндри обладнують вбудованими безконтактними індуктивними датчиками переміщення, які дають змогу фіксувати та передавати в електронну систему керування сигнал про фактичне положення поршня. Вбудовані в сервоциліндр ущільнення не навантажуються робочим тиском, що дає змогу мінімізувати тертя, яке в 3...4 рази менше порівняно зі звичайними гідроциліндрами. Для гідроциліндрів, які працюють в широкому діапазоні швидкостей за великих бічних навантажень на штоку, застосовують гідростатичні «кишені» опори штока (рис. 2.23). Такі циліндри використовують для робочого тиску до 28 МПа і номінальних зусиль від 10 до 10000 кН. Гідростатична опора такого сервоциліндра має чотири розміщені навколо штока кишені, які центрують шток всередині наявного радіального зазору. Коли немає радіальної сили, тиск в кишенях дорівнює 50% робочого тиску p . Радіальна сила викликає деяке радіальне зміщення штока, через що зростає тиск в кишені, від якого залежить центрування штока. Для отримання високих динамічних властивостей гідроприводів довжина з'єднувальних гідроліній між системою керування та сервоциліндром повинна бути мінімальною. Для того щоби домогтися цього, сервоблок встановлюють безпосередньо на сервоциліндрі. Трубопроводи під'єднують до агрегата через сервоблок, який має системи обмеження зусилля, фільтрації робочої рідини, що надходить до опорних підшипників та реалізує акумулювання гідравлічної енергії.

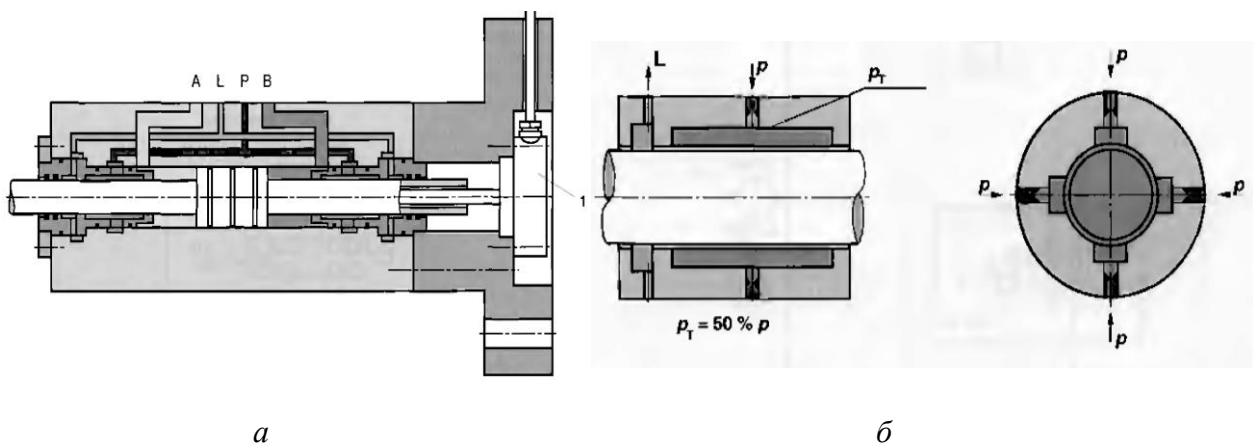


Рис. 2.23. Схеми сервоциліндра з гідростатичними «кишенями» опори штока (а); опори штока (б)

Швидкість переміщення поршня визначають за відомою формулою

$$V = \frac{Q}{S}, \quad (2.10)$$

де S – площа поршня; Q – подача рідини.

Площа поршня у поршневій порожнині:

$$S = \frac{\pi D^2}{4}, \quad (2.11)$$

де D – діаметр поршня.

Площа поршня у штоковій порожнині:

$$S = \frac{\pi}{4} (D^2 - d^2), \quad (2.12)$$

де d – діаметр штока поршня.

Сила на штоку гідроциліндра за швидкості поршня V і тиску p :

$$F = p \frac{Q}{V} = pS. \quad (2.13)$$

Теоретична потужність гідроциліндра:

$$N_T = FV. \quad (2.14)$$

Час переміщення поршня в гідроциліндрі на відстань h :

$$t = \frac{Sh}{Q}, \quad (2.15)$$

де h – відстань переміщення поршня.

Інтелектуальний мехатронний модуль – це конструктивно та функціонально самостійний виріб з інтегрованими механічною, електричною, інформаційною та комп'ютерною (електронною) частинами, який можна використовувати окремо та в різних комбінаціях з іншими модулями. Порівняно з мехатронними модулями руху, в конструкції інтелектуальних модулів руху додатково вбудовують мікропроцесорні обчислювальні пристрої та силові електронні перетворювачі. На рис. 2.24 зображено інтелектуальні мехатронні модулі фірми Siemens.

Перевагами інтелектуальних мехатронних модулів руху є такі:

- здатність виконувати складні рухи самостійно, без звернення до верхнього рівня керування, що підвищує автономність і гнучкість для мінливих та невизначених умовах зовнішнього середовища;
- спрощення комунікацій між модулями і центральним пристроєм керування (з переходом до бездротових комунікацій), що підвищує точність та здатність до швидкої зміни конфігурації;

- підвищення надійності та безпеки мехатронних систем завдяки комп'ютерній діагностиці несправності та автоматичного захисту в аварійних та позаштатних режимах роботи;
- створення розподілених систем керування із застосуванням мережних методів, апаратно-програмних платформ на базі персональних комп'ютерів та відповідного програмного забезпечення;
- використання сучасних методів теорії управління (програмних, адаптивних, інтелектуальних, оптимальних) безпосередньо на виконавчому рівні, що істотно підвищує якість конкретних процесів керування в конкретних реалізаціях;
- інтелектуалізація силових перетворювачів, що належать до складу інтелектуальних модулів руху, для виконання безпосередньо в мехатронному модулі інтелектуальних функцій з керування рухом, захисту модуля в аварійних режимах та діагностики несправностей;
- інтелектуалізація сенсорів для мехатронних модулів дає змогу досягти більш високої точності вимірювання, програмним шляхом домігшись у сенсорному модулі фільтрації шумів, калібрування, лінеаризації характеристик «вхід/вихід», компенсації перехресних зв'язків, гістерезису і дрейфу нуля.

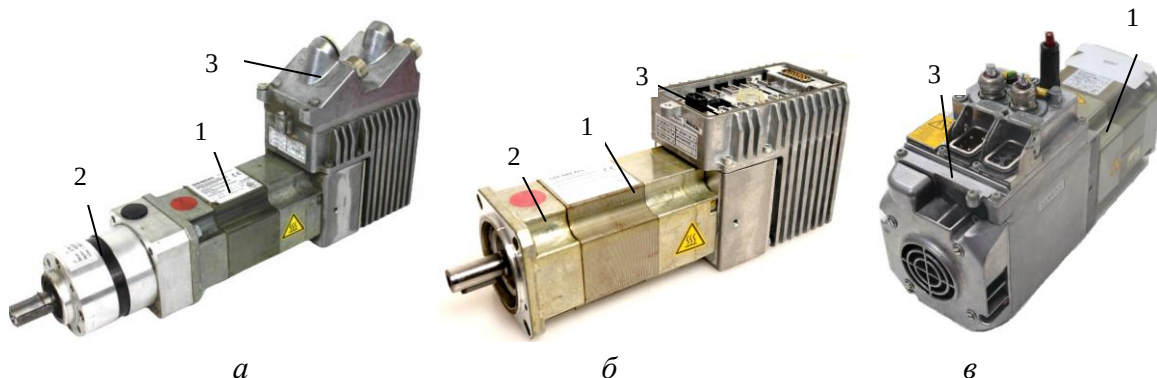


Рис. 2.24. Інтелектуальні мехатронні модулі Siemens: *а* – Simodrive Posmo A 6SN2132; *б* – Simodrive Posmo A 300W 6SN2155; *в* – Simodrive Posmo Si 6SN2460; 1 – електродвигун; 2 – механічний перетворювач (редуктор); 3 – система електронного керування

Основним фактором, що стримує використання інтелектуальних мехатронних модулів у серійних виробках, є їхня висока ціна, проте згодом вартість таких виробів буде знижуватися, що пов'язане з низкою таких чинників:

- стрімким розвитком апаратних пристроїв та інформаційних технологій, орієнтованих на завдання з керування рухом;

- появою напівпровідникових приладів (силових польових транзисторів, біполярних транзисторів з ізолюваним затвором, тиристорів з польовим керуванням);
- переходом на нову елементну базу в системах керування рухом – це цифрові сигнальні процесори (DSP-процесори) та блоки FPGA (Field Programmable Gate Arrays);
- розробкою гібридних технологій, що дозволяють вбудовувати електронні та обчислювальні пристрої в механічні вузли (як приклад підшипники з вбудованими датчиками (рис. 2.25)).



Рис. 2.25. Підшипники з вбудованими датчиками

Підшипники, що мають вбудований датчик, який збирає інформацію про температуру, вібрацію та швидкість обертання підшипника і передає її бездротовою мережею. Такі підшипники можуть бути цілком автономними і без кабелів для підзарядки та передачі даних, оскільки ці дані можуть бути передані бездротовою мережею, а датчик та радіомодуль живляться електроенергією, виробленою обертанням підшипника. Датчики, вбудовані у зовнішнє кільце підшипника, максимально наближені до поверхні доріжки кочення, що робить їх точним та чутливим до найменших проявів ушкоджень у підшипнику. Підшипники із вбудованими датчиками мають такі самі габарити і характеристики вантажопідйомності, як і стандартні підшипники. Завдяки цьому вони взаємозамінні зі звичайними підшипниками.

Інтелектуальний мехатронний модуль складається з таких основних елементів: електродвигуна (можливим є використання рушіїв інших типів, наприклад, гідравлічних); механічного перетворювача; датчиків

зворотного зв'язку та сенсорних пристроїв; керівного контролера; силової електроніки; пристроїв сполучення та зв'язку. В сучасних інтелектуальних мехатронних модулях використовують різні типи електродвигунів: обертові та лінійні, змінного та постійного струму, колекторні та вентильні, безперервного руху та крокові. В ролі перетворювачів руху застосовують зубчасті, гвинтові та інші передачі. Якщо інтелектуальний мехатронний модуль побудовано з використанням високомоментних двигунів, перетворювачів руху не застосовують. Вбудовані керівні контролери на сучасній елементній базі дають змогу отримувати компактні та надійні мехатронні вироби, що мають інтелектуальні функції, і будувати на їхній основі багатокоординатні мехатронні системи з децентралізованим управлінням. У поєднанні з відкритою архітектурою систем керування типу РС-НС можна створити якісно нові системи керування, що мають високу швидкодію, точність та функціональну гнучкість.

В інтелектуальних мехатронних модулях використовують різні датчики положення і швидкості (фотоімпульсні трансформатори, що обертаються, тахогенератори, датчики струму і моменту, температури і вібрації), які передають інформацію в пристрій комп'ютерного керування про фактичний стан підсистем модуля.

Принципово важливо, щоб усі ці елементи були конструктивно об'єднані розробником в єдиному корпусі. При цьому пристрої зв'язку стають внутрішніми блоками, недоступними для користувача.

Розглянемо загальну структуру однокоординатного мехатронного модуля, що наведений на рис. 2.26.

Пристрій комп'ютерного керування має два вхідні інформаційні канали: інтерфейс *I1* пов'язує його з верхнім рівнем керування, а інтерфейс *I4* – з сенсорами й одним вихідним каналом (інтерфейс *I2*), через який надходять керівні команди на виконавчий привід.

З наведеної схеми можна виділити три напрями інтелектуалізації мехатронних модулів руху, які класифікуються залежно від інтерфейсних точок інтеграції.

1. Розвиток інтегрованих інтерфейсів, що пов'язують керівний контролер з комп'ютером верхнього рівня в єдиний апаратно-програмний керівний комплекс (інтерфейс *I1*).

2. Створення інтелектуальних силових модулів керування рухом інтеграції керівних контролерів і силових перетворювачів (інтерфейс *I2*).

3. Розроблення інтелектуальних сенсорів, які додатково до звичайних вимірювальних функцій виконують комп'ютерну обробку і перетворення сигналів за гнучкими програмами (інтерфейс І3).

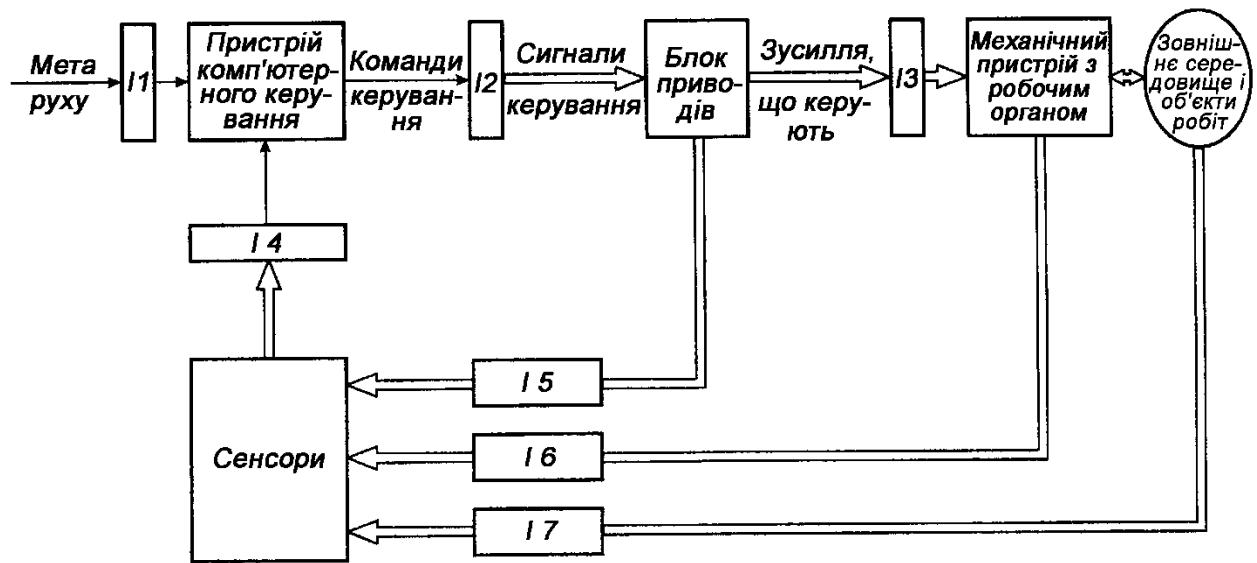


Рис. 2.26. Блок-схема традиційної машини з комп'ютерним керуванням:
І1–І7 – інтерфейсні блоки

Розробляючи ІММ, слід послідовно виконати етапи функціонально-структурного та структурно-конструктивного аналізу і потім розпочати конструкторське втілення обраного варіанта. Слід зазначити, що сутність інтелектуальних мехатронних модулів визначається не типом структурних елементів, використаних у цій конструкції, а насамперед їх функціональним призначенням.

Контрольні запитання

1. Які типи електродвигунів вам відомі?
2. Як керують електродвигуном постійного струму?
3. Як керують безщітковим електродвигуном постійного струму?
4. Що являє собою кроковий електродвигун?
5. Що називають сервоприводом?
6. Як керувати сервоприводом?
7. Як керують кроковим двигуном?
8. Які види крокових двигунів вам відомі?
9. Як керувати асинхронним двигуном змінного струму?
10. Що таке частотний перетворювач?
11. Назвіть види керування у частотних перетворювачах.

Лекція 3. Принципи побудови мехатронних структур

Особливість підходу до проектування мехатронних структур полягає в інтеграції в єдиний функціональний модуль двох або більше елементів, можливо навіть різної фізичної природи (рис. 3.1). На відміну від традиційної автоматизованої системи приводу в мехатронних системах важливе значення для функціонування сукупної системи має взаємодія механічних й електронно-цифрових компонентів, тому мехатронна система відрізняється тим, що, починаючи з концепції, досягається просторово і функціонально інтегрована сукупна система.

Мехатронний модуль в ідеальному для користувача варіанті, отримавши на вхід інформацію про мету керування, повинен виконувати заданий функціональний рух з допустимою похибкою. Це означає, що апаратне об'єднання елементів в єдині конструктивні модулі повинно також обов'язково супроводжуватися розробкою інтегрованого програмного забезпечення. Програмні засоби потрібні для безпосереднього переходу від задуму системи через її математичне моделювання до керування функціональним рухом у реальному часі.

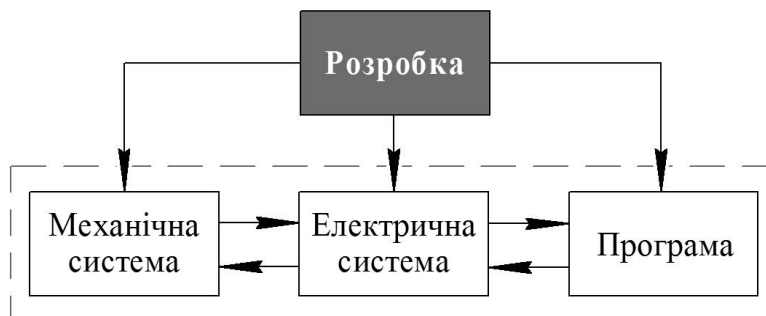


Рис. 3.1. Підхід до розроблення мехатронних систем

Відповідно до ієрархії рівнів інтеграції в мехатронних системах виділяють перший рівень розроблення, який пов'язаний із синтезом примітивних мехатронних пристроїв та їхніх складових елементів. Другий рівень полягає в розробленні інтегрованих мехатронних модулів. На третьому рівні розробляють багатокоординатні мехатронні машини, які компонують з інтегрованих мехатронних модулів. Вищий (четвертий) рівень припускає розроблення на єдиній інтеграційній платформі комплексів мехатронних машин для реконфігурованого виробництва. У розробленні мехатронних систем послуговуються також VDI-

директивою, яка визначає принципові дії послідовних кроків з розроблення мехатронних систем (рис. 3.2), зокрема:

- визначення вимог щодо конкретної розробки;
- розроблення концепції рішення, що охоплює всі галузі знань, які пов'язані з визначенням відповідних підфункцій;
- специфічні розрахунки для конкретної галузі знань, зокрема, критичних компонентів системи;
- інтеграція систем, тобто інтеграція окремих галузей знань у сукупну систему.

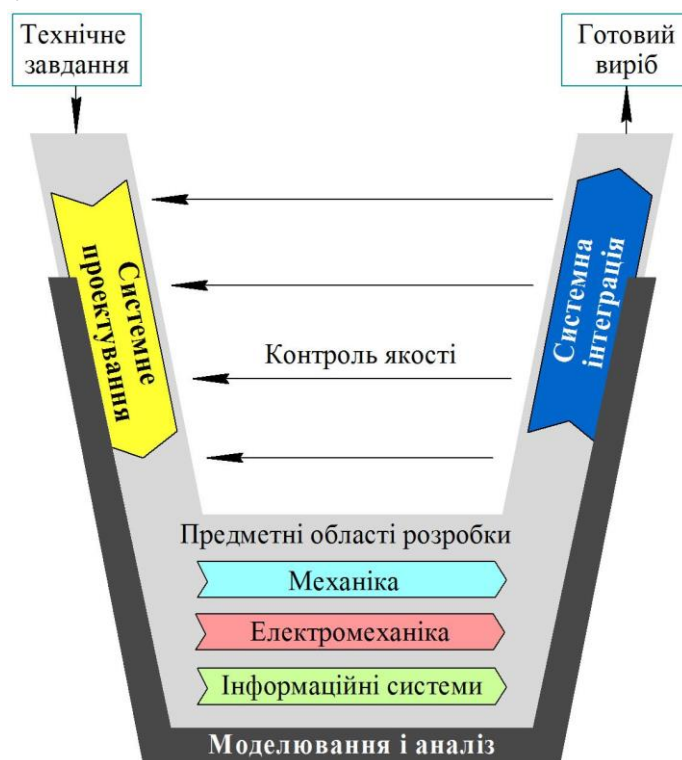


Рис. 3.2. V-модель циклу розроблення

У виборі та втілення концепції мехатронної системи важливу роль відіграють функціонально та формально орієнтовані моделі.

Функціонально орієнтовані моделі містять базові (кінематичні, кінетичні і мехатронні) функції, вони призначені для опису характеристик мехатронних систем. Як відповідний клас моделей для розроблення різноманітних мехатронних проблем зарекомендували себе регульовані багатокорпусні системи (MKS). У найпростішому випадку під цим розуміють відкритий кінематичний ланцюг жорстких тіл, які пов'язані одне з одним за допомогою шарнірів і на рух яких можна впливати за допомогою виконавчих сил і моментів. Завдяки MKS можна досягти реалістичного моделювання системи.

Формально орієнтовані моделі застосовують для розрахунку міцності та конструктивної розробки підсистем сукупної мехатронної системи. Для цього використовують CAD та CAE програми та відповідні зв'язки обох програм для реалістичного опису геометрії і властивостей міцності. Процес розроблення відбувається шляхом циклічного послідовного застосування функціонально та формально орієнтованих моделей з відповідними інструментами (MKS, FEM, CAD) для їхнього дослідження.

Проектування мехатронних систем є складною багатфакторною проблемою вибору й оптимізації ухвалених технічних, технологічних, організаційно-економічних та інформаційних рішень. Вузким місцем мехатронних модулів і машин є інтерфейси між пристроями й елементами. Поняття «інтерфейс» є ключовим для пропонованого підходу до проектування мехатронних модулів і систем. Насамперед відзначимо, що взаємодія основних пристроїв у мехатронній системі відбувається не безпосередньо, а через певні сполучні блоки. Місце інтерфейсу в структурі мехатронної системи задається зв'язками із вхідними й вихідними пристроями. Технічні характеристики інтерфейсу визначаються способом і процедурою передавання (за потреби – перетворення, зберігання й синхронізації) впливів, сигналів та інформації, а також апаратно-програмною реалізацією використовуваних каналів зв'язку.

Надійність, точність виконання рухів, стійкість до несприятливих зовнішніх впливів, модульність й компактність конструкції відіграють також важливу роль у проектуванні інтегрованих мехатронних машин. З погляду споживача, це повинні бути цілісні вироби, зручні у налаштуванні й програмуванні рухів. Оскільки тип мехатронного модуля визначається набором мехатронних пристроїв, до яких належать дві основні групи: у виконавчу групу входять механічні пристрої й двигуни, а група інтелектуальних пристроїв є сукупністю електронних, інформаційних і керівних компонентів, тому в розробленні слід застосовувати підхід системного проектування.

Системи з глибоким ступенем інтеграції елементів мають і зворотний бік – такі машини є менш гнучкими (мають обмежені можливості для модернізації й реконфігурації). Тому мехатронні модулі різного рівня інтеграції дають змогу виявляти раціональну структуру під конкретні завдання з автоматизації. Переваги комбінованої системи –

багатофункціональність, компактність, економічність, зручність здатні також перетворити таку систему на негнучку, окремі її компоненти не можна замінити й модернізувати, їх можна експлуатувати й ремонтувати тільки разом.

У проєктуванні складних систем застосовують такі напрями теорії системного проєктування:

- функціонально-структурний аналіз і еволюційний синтез технічних систем;
- методологію паралельного проєктування систем;
- структурний синтез і оптимізацію технічних систем за критеріями складності.

Функціонально-структурний підхід ґрунтується на ідеї пріоритету функції системи над її структурною організацією. Завданням проєктування є визначення такої структури, яка дасть системі змогу виконувати функціональні завдання з максимальною ефективністю за обраними критеріями якості. Основні положення цього методу розроблено стосовно завдань з проєктування інформаційно-керівних комплексів й обчислювальних пристроїв.

Метод еволюційного синтезу означає пошук раціональних рішень шляхом багатоетапної процедури оптимізації. Пошук варіантів виконують з бази перспективних структурних рішень, яка відкрита для редагування (удосконалення) і має ієрархічну структуру. Такий метод проєктування дає змогу підвищити якість проєкту й знизити вимоги до досвіду розроблювача.

Методологічною основою для розроблення мехатронних систем слугують методи паралельного проєктування. У традиційному проєктуванні розроблення механічної, електронної, інформаційної й комп'ютерної частин ведеться послідовно й незалежно одна від одної (рис. 3.3).



Рис. 3.3. Традиційний алгоритм проєктування

Методологія паралельного проектування полягає в одночасному й взаємозалежному синтезі всіх пристроїв мехатронної системи. Одна з відомих процедур проектування інтегрованих мехатронних машин представлена на рис. 3.4.



Рис. 3.4. Алгоритм паралельного проектування мехатронних систем

Процедура паралельного проектування являє собою чотири взаємозалежні етапи (рис. 3.5):

- визначення функцій мехатронних модулів на підставі аналізу вихідних вимог до мехатронної машини;
- функціонально-структурний аналіз з метою вибору й формування структури всіх мехатронних модулів системи;
- структурно-конструктивний аналіз, конструювання й формування моделі модулів системи;
- планування й оптимізація функціональних рухів, розроблення програм руху машини та її модулів.

Вихідні вимоги до мехатронних машин, що визначають основні проєктні рішення, формуються шляхом аналізу виконуваних технологічних процедур. Рекомендації технологів, як правило, обмежуються вибором номінальних режимів руху машини. Проектуючи мехатронну систему, зазвичай задають вихідні дані, що містять інформацію про програмувальні рухи й комплекси обмежень. Вимоги до траєкторій руху містять опис програмувальних траєкторій робочого органа в просторі й припустиму похибку. Траєкторії робочого органа визначаються геометрією контурів і поверхонь об'єктів робіт. На наступному етапі проектування формується структура мехатронних модулів зі схемою енергетичних й інформаційних потоків. Наступний етап призначений для синтезу модулів і мехатронної машини в цілому. Усі проєктні етапи мають циклічний характер.



Рис. 3.5. Процедура проєктування інтегрованих мехатронних машин

Побудова функціональної, структурної й конструктивної моделей дає змогу застосовувати в мехатроніці методи й засоби автоматизованого аналізу, проєктування й конструювання.

У розробленні інтегрованих мехатронних модулів можуть бути застосовані три основні методи, які можна класифікувати за характером об'єднання пристроїв і розв'язання «проблеми інтерфейсу» в мехатронних системах. Кожен з цих методів може бути використаний окремо або в комбінації з іншими на різних етапах проєктування.

За першим методом інтегровані машини створюють шляхом вилучення проміжних перетворювачів і відповідних інтерфейсів з їхньої структури. Це найбільш глибокий рівень інтеграції, коли завдяки відмові від багатоступеневого перетворення енергії й інформації у мехатронних системах створюють фундаментальну основу для досягнення високої точності, швидкодії, компактності та надійності. У разі застосування методу вилучення проміжних перетворювачів та інтерфейсів систему розробляють так, щоб уникнути використання проміжних елементів, які зазвичай слугують для перетворення сигналів, енергії або інформації між різними компонентами системи. Замість цього створюють конструкцію, у якій пристрої і механізми взаємодіють безпосередньо між собою, обмежуючи використання додаткових проміжних шарів або перетворювачів. Метод застосовують зазвичай на другому етапі проєктування мехатронних систем, синтезуючи структуру модулів.

Застосування методу просте. Припустимо, є мехатронний пристрій, який складається з трьох основних компонентів: сенсора (S), приводу (М) і контролера (С). У звичайній системі без використання методу вилучення проміжних перетворювачів кожен компонент може бути пов'язаний з проміжними елементами для забезпечення взаємодії та обміну інформацією. Застосовуючи цей метод, можна досягти безпосередньої взаємодії між компонентами, уникнувши використання проміжних елементів (рис. 3.6).

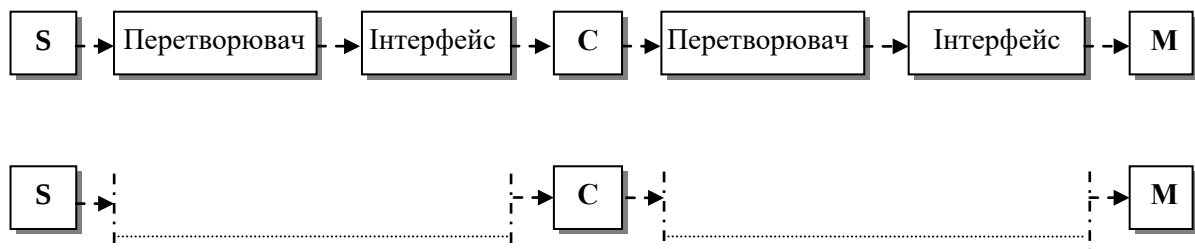


Рис. 3.6. Принцип застосування методу вилучення проміжних перетворювачів і відповідних інтерфейсів

Ідея полягає в тому, що безпосередня комунікація між компонентами може поліпшити ефективність та характеристики системи. Проте такий підхід ускладнює функції основних компонентів, оскільки на них переносяться відповідальність інтерфейсів та перетворювачів.

Другий метод означає апаратно-конструктивне об'єднання пристроїв різної фізичної природи в одному корпусі багатофункціонального мехатронного модуля (рис. 3.7).

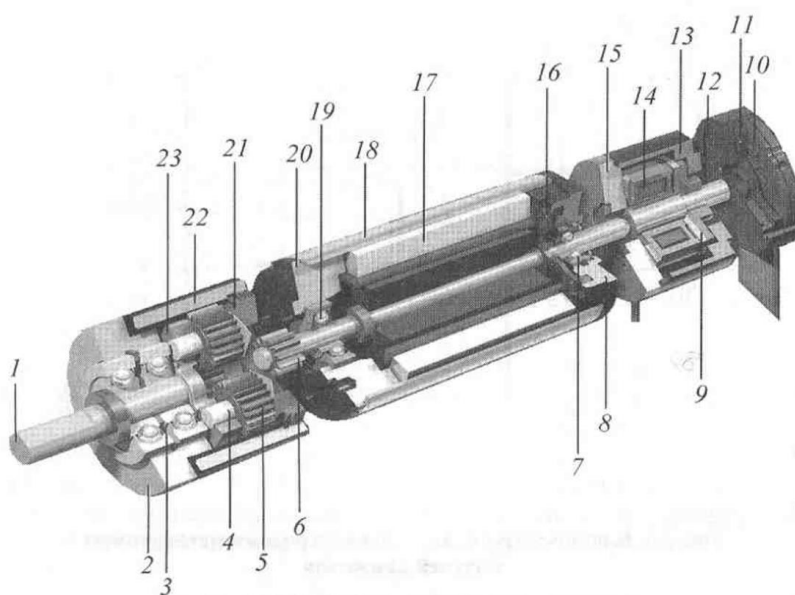


Рис. 3.7. Мехатронний модуль ЕС-тах Махон:
1 – вихідний вал;
2 – фланець редуктора;
3, 7, 19 – підшипники;
4 – вісь; 5 – сателіт планетарної передачі;
6, 21 – зубчасті колеса;
8, 20 – фланець двигуна;
9 – гальмівний диск;
10 – плата керування;
11 – кодовий диск енкадера;
12 – корпус датчика;
13 – корпус гальма;
14 – статор;
15 – кришка;
16 – датчик Хола;
17 – статор електродвигуна;
18 – корпус електродвигуна;
22 – корпус редуктора;
23 – водило

Виготовлені за цієї методикою модулі руху являють собою єдині вироби, підвищується їхня ефективність, спрощується процес монтажу та обслуговування, покращується масштабованість. Цей підхід може бути ефективним у випадках, коли треба створити компактні, готові до використання модулі з високим рівнем інтеграції для виконання конкретних завдань або функцій в мехатронних системах. Вдалим прикладом є модуль ЕС-тах фірми Махон, яка випускає мінісистеми діаметром до 40 мм, завдовжки від 24 до 90 мм та потужністю усього 120 Вт (рис. 3.7), і при цьому в єдиному корпусі розміщуються безконтактний електродвигун, планетарний редуктор, гальмо, датчик та плата керування. Технологічною базою цього методу інтеграції є гібридне складання вузлів й елементів. Апаратне й конструктивне об'єднання елементів у єдині модулі повинно супроводжуватися розробленням інтегрованого програмного забезпечення. Іншими прикладами модулів руху, створених за таким методом, можуть бути мотор-редуктор, мотор-колесо (рис. 3.8), шпиндель-мотор і мотор-барабан. До складу сучасних модулів руху окрім двигунів і перетворювачів руху належать інші види механічних пристроїв – гальма й люфтовибирачі, напрямні і перетворювачі руху.

Третій метод полягає в перенесенні функціонального навантаження від механічних вузлів до інтелектуальних (електронних і програмних). Цей метод дає змогу досягти поступового зниження собівартості пристроїв шляхом мінімізації механічної складності мехатронної системи за розширення її функціональних можливостей.

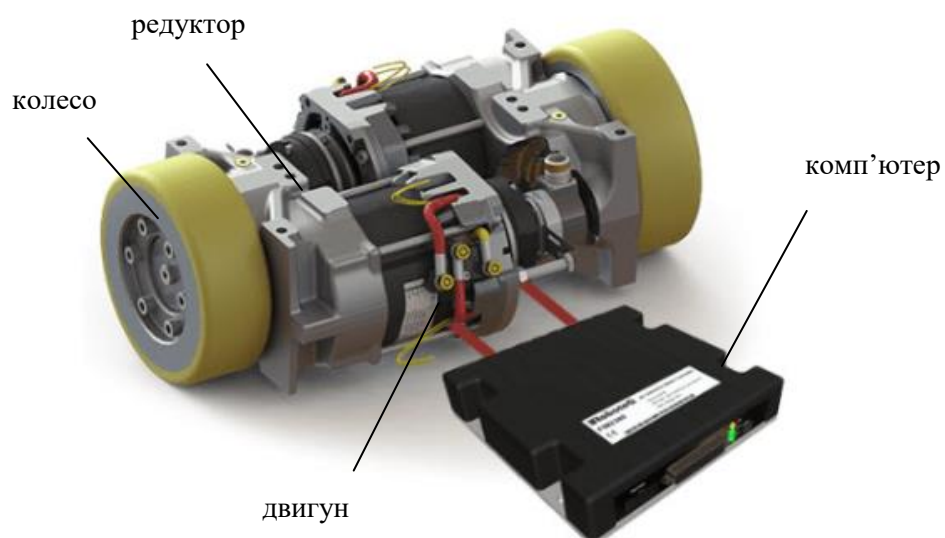


Рис. 3.8. Мехатронний модуль привідної підвіски робота KuKa з мотор-колесами

Основні характеристики методу включають:

- *перенесення функціональності* означає, що мехатронними системами замінюють традиційні механічні рішення із залученням інтелектуальних систем. Наприклад, замість використання механічних елементів для виконання певних завдань, таких як керування рухом або обробка сигналів, функції переносяться на електронні компоненти;
- *використання інтелектуальних технологій* означає, що інтелектуальні пристрої, такі як мікроконтролери, сенсори, програмне забезпечення штучного інтелекту тощо, застосовують для вирішення завдань, які традиційно виконували за допомогою механічних систем. Це може бути керування рухом, обробка даних, аналіз сигналів, ухвалення рішень;
- *підвищення гнучкості та адаптивності* дає змогу створювати більш гнучкі й адаптивні мехатронні системи, оскільки функціональність може бути змінена чи адаптована шляхом перепрограмування інтелектуальних пристроїв. Це корисно в умовах змінних вимог або зміни середовища;
- *оптимізація ресурсів* відбувається, коли перенесення функціональності на інтелектуальні пристрої сприяє більш ефективному використанню ресурсів, таких як енергія, простір й обчислювальна потужність. Інтелектуальні пристрої здатні ефективно вирішувати завдання за допомогою алгоритмів та програм, отже, досягати оптимальності у функціонуванні системи.

У мехатронних системах цей метод можна застосовувати до елементів керування, коли інтелектуальні контролери аналізують сигнали з сенсорів та вирішують, яким чином краще керувати рухом чи виконувати інші завдання, традиційно виконувані механічними системами.

Відомі два основні підходи для побудови моделей складних технічних систем. Перший полягає у функціональному визначенні розглянутої системи через її поведінку стосовно зовнішніх об'єктів і зовнішнього середовища. Другий підхід оснований на структурному представленні системи і зв'язків між її елементами. Кожний зі способів побудови складних систем задає деяку структуру елементів, які досягають поставленої мети різним способом виконання функцій або різною структурою системи.

Розрізняють функціональну та структурну організацію складних систем. Під функціональною організацією розуміють спосіб побудови

функціональності системи, достатньої для виконання поставленої мети. Структурна організація означає спосіб синтезу потрібних елементів системи для виконання поставлених функцій.

Досить часто функціональне представлення мехатронного модуля відображається у формі «абстрактної скриньки» (рис. 3.9), яка містить два інформаційні входи (програма руху та інформаційний зворотний зв'язок), додатковий механічний вхід (сили реакції зовнішнього середовища) і один вихід – цілеспрямований механічний рух. Отже, у загальному випадку функціональна схема мехатронного модуля може бути побудована як інформаційно-механічний перетворювач.

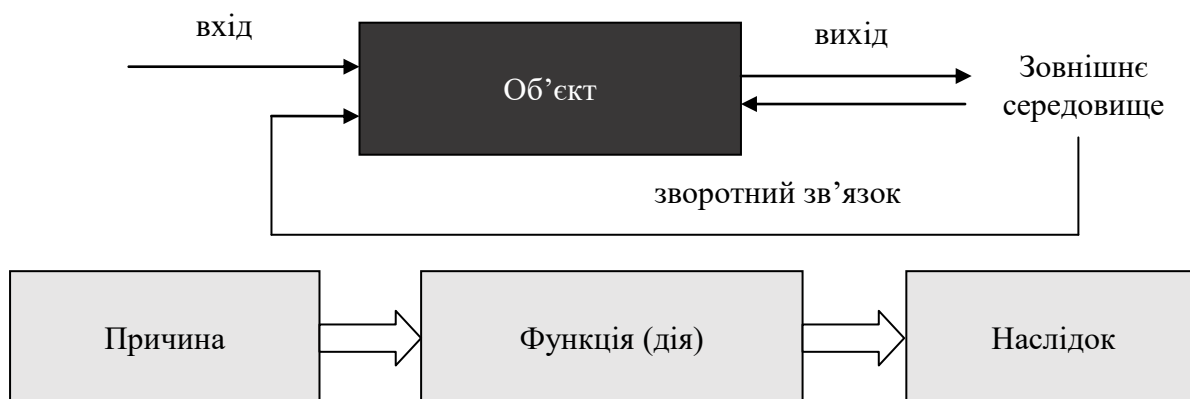


Рис. 3.9. Функціональне представлення мехатронної системи

Вхід такої «абстрактної скриньки» завжди є причиною до зміни її стану та виконання дії на виході, що змінює зовнішнє середовище та породжує наслідок. Таке абстрактне представлення мехатронної системи зумовлює кілька висновків:

- стан системи однозначно в деякий момент часу визначає вихід системи;
- попередній стан системи в певний момент часу визначають її майбутні стани;
- в один проміжок часу система не може перебувати в різних станах.

Для опису станів технічних систем застосовують різні математичні моделі.

Структурна модель мехатронного модуля відображає склад його елементів і зв'язки між ними. Структурні моделі можна графічно представити у вигляді блок-схем.

Головною особливістю сучасного етапу розвитку мехатроніки є створення принципово нового покоління модулів – інтелектуальних. На

відміну від мехатронних модулів руху, в їхню конструкцію додатково вбудовуються комп'ютерні пристрої й силові електронні перетворювачі, що надає цим модулям інтелектуальних властивостей та є їхньою головною відмітною ознакою. До складу сучасних мехатронних модулів належать також інтелектуальні силові перетворювачі, що поєднують електричне й електроінформаційне перетворення. Їхня особливість полягає в тому, що вони містять вбудовані блоки мікроелектроніки, які призначені для виконання інтелектуальних функцій – керування рухом, захист в аварійних режимах і діагностика несправностей. Інтелектуальні силові перетворювачі будують на основі напівпровідникових приладів. Типовими представниками цих приладів є силові польові транзистори (MOSFET), біполярні транзистори з ізольованим затвором (IGBT) та замкнуті тиристори з польовим керуванням (MCT).

У процесі розроблення мехатронних систем важливо мати математичні моделі для аналізу статичної та динамічної поведінки. Ці моделі потрібні для виконання симуляцій, проектування систем керування та відновлення значень змінних. Основними способами отримання таких моделей є теоретичне моделювання на основі фізичних принципів та експериментальне моделювання з використанням вимірюваних величин.

Основним засобом опису мехатронних систем є використання диференціального й інтегрального числення, цифрових методів. Аналіз мехатронних систем допускає послідовність їхнього узагальненого, графічного, аналітичного й алгоритмічного опису, а їхній аналіз і синтез пов'язані з узагальненням і розробленням єдиного математичного апарату алгоритмізації процесів взаємодії елементів систем.

У теоретичному моделюванні потрібно вміти узагальнювати фізичні процеси, оскільки компоненти мехатронних систем мають різну предметну сферу. Актуальним в цьому є принципи теоретичного моделювання за потоком енергії, які дають змогу уніфікувати компоненти в різних процесах, таких як електротехніка, механіка та теплотехніка. На рис. 3.10 відображено структурний підхід на основі потоків із застосуванням бондграфів зв'язку.

Бондграф – графічне подання динамічної системи в описі тієї чи іншої фізичної (механічної, електричної, гідравлічної, пневматичної, економічної тощо) системи, що відображає процес перерозподілу енергії в цій системі. Бондграф оснований на законах збереження енергії.

Основна відмінність від блок-схем або графів проходження сигналів полягає в тому, що в бондграфі ребрам ставиться у відповідність потік енергії, який може бути спрямований в обидва боки, у той час як у блок-схемах та графах проходження сигналів являє собою односпрямований потік інформації. Ребра в бонд-графах оснащують символами, що задають або потік енергії, або потік інформації.

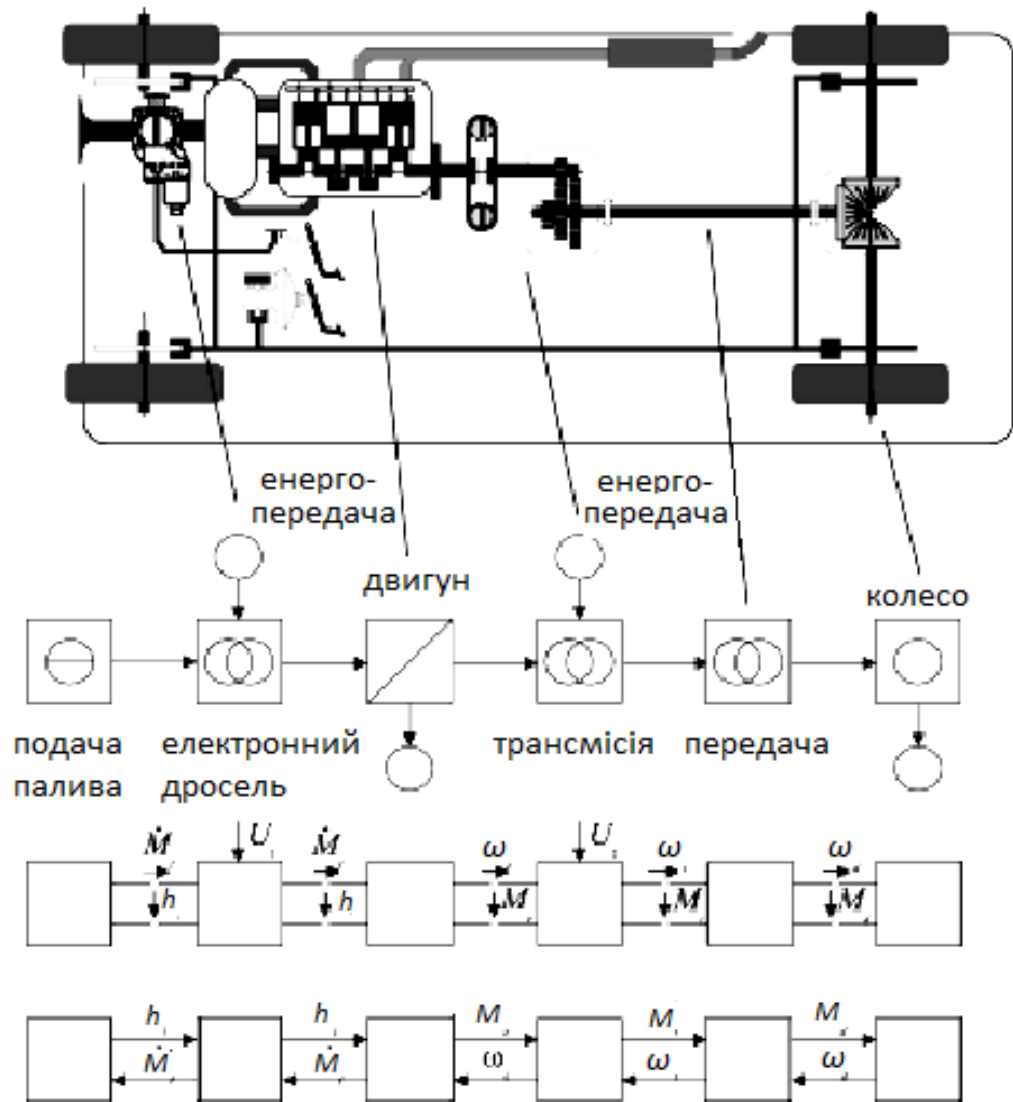


Рис. 3.10. Функціональне представлення мехатронної системи автомобіля у вигляді графів зв'язку

Порівняно з іншими засобами візуального подання типу блок-схем бондграфи мають багато переваг:

- у них розрізняють потоки енергії та потоки інформації;
- оскільки бондграфи спираються на закон збереження енергії, виявляється неможливим ввести на розгляд енергію, якої немає в системі;

- виділяють причинні зв'язки між зусиллями (сила, напруга, тиск) і потоками (швидкість, струм, витрата). Такі причинні зв'язки задаються один раз, коли створюється вихідна схема, що дає змогу виявити також модельовані явища, наприклад, такі як струм, кутова швидкість;

- оскільки кожен зв'язок представляє потік в обох напрямках, в системах з протидією, наприклад, з електрорушійною силою немає потреби в додаткових зв'язках для опису впливу елемента самого на себе.

Якщо динаміка модельованої системи характеризується різними часовими масштабами, швидкі процеси в реальному часі можуть бути розглянуті як миттєві явища за допомогою гібридних бондграфів.

У бондграфі розрізняють:

- вузли (вершини), яким відповідають «фізичні явища», описувані рівняннями. Це загальне поняття може означати механічні деталі, електричні складові, гідравлічні пристрої тощо. Вузлу може бути відповідна і підмножина деталей, іншими словами, вузол сам по собі може бути описаний як вкладений бондграф. Але в той же час фізичний закон застосовується до системи загалом (наприклад, правила Кірхгофа для електричних ланцюгів);

- дуги (ребра), яким відповідні потоки енергії. Інакше кажучи, вони визначають дію одного вузла на інший. Їх називають «зв'язками» (бондами), звідки й походить назва графа.

Графи зв'язків дають високорівневу абстракцію, мають невелику гнучкість та потребують додаткових зусиль для генерації алгоритмів симуляції.

Морфологічний аналіз є методом системного аналізу, який застосовують для дослідження структури і можливих конфігурацій складних технічних систем. Основна ідея полягає в поділі системи на елементарні компоненти чи параметри, або «морфеми», які можуть бути комбіновані для створення різних варіантів системи (табл. 3.1).

На першому етапі виконують аналіз та розбір системи на основні функціональні елементи, параметри, аспекти та інші компоненти, які можуть визначати структуру системи. Для кожного елемента або параметра створюють файл, в якому вказують можливі варіації чи альтернативи цього елемента. Таким чином, для кожної морфеми визначають різні варіації. За допомогою комбінаторних методів об'єднують різні варіанти морфем, утворюючи нові конфігурації

системи. Далі аналізують кожний варіант з погляду його можливостей, вартості, ефективності та інших критеріїв.

Морфологічний синтез – це етап, коли на підставі результатів аналізу обирають оптимальні варіанти для подальшого розроблення технічної системи. Аналізуючи можливі варіанти, визначають оптимальні комбінації морфем відповідно до поставлених завдань і критеріїв. Обрані комбінації перетворюють в концепцію технічної системи, що охоплює її структуру, функції, параметри та інші ключові характеристики.

Таблиця 3.1

Приклад морфологічної таблиці

Функції елементів системи					
	1	2	3	4	5
	функція 1	функція 2	функція 3	функція 4	функція 5
1	рішення 1.1	рішення 2.1	рішення 3.1	рішення 4.1	рішення 5.1
2	рішення 1.2	рішення 2.2		рішення 4.2	рішення 5.2
3	рішення 1.3				рішення 5.3
4	рішення 1.4				
5	рішення 1.5				

Гнучка методологія розроблення (agile development) – це клас методологій, що ґрунтується на ітеративній розробці, в якій вимоги та розв'язки еволюціонують завдяки співпраці між багатофункціональними командами.

Більшість гнучких методологій націлені на мінімізацію ризиків шляхом зведення розроблення до серії коротких циклів, що мають назву ітерацій, які зазвичай тривають один-два тижні. Кожна ітерація сама собою має вигляд програмного проєкту в мініатюрі, охоплюючи всі завдання, потрібні для мінімального приросту за функціональністю. Після закінчення кожної ітерації розробник виконує переоцінку пріоритетів розроблення.

Гнучке розроблення для мехатронних систем дає змогу ефективно відображати складну взаємодію між механічними, електронними та програмними компонентами, досягаючи гнучкості та адаптивності на всіх етапах життєвого циклу системи.

Real-Time Simulation (симуляція в реальному часі) є важливим аспектом розроблення мехатронних систем, оскільки дає змогу

знаходити рішення та валідувати роботу системи в умовах, близьких до реальних, не вдаючись до фізичного виготовлення пристрою. Завдяки такому підходу можна ефективно визначати характеристики та функції мехатронної системи ще до її фізичного втілення.

Основні переваги Real-Time Simulation такі:

- симуляція в реальному часі, що дає змогу брати до уваги реальні умови роботи мехатронних систем, зокрема взаємодію різних компонентів, сенсорів, приводів та керування;
- завдяки Real-Time Simulation можна тестувати та валідувати різні сценарії роботи системи, виявляти можливі проблеми та уточнювати параметри ще на етапі віртуального моделювання;
- зменшення часу між розробленням та фізичним виготовленням;
- апаратне забезпечення дає змогу досягти високої швидкості та точності відтворення реальних умов;
- тестування на межах функціональності виявляє проблеми, які можуть виникнути в реальних умовах експлуатації.

Проектування мехатронних модулів ґрунтується на спільному аналізі їхньої функціональної, структурної й конструктивної моделей. Метою проектування мехатронного модуля є перетворення вихідних вимог у конструкторську реалізацію й відповідну документацію, за якою може бути виготовлена система, що задовольняє сформульовані показники якості. На рис. 3.11 подано загальний алгоритм проектування мехатронного модуля, який складається з трьох основних етапів проектування – це послідовно виконуваний функціональний, структурний і конструктивний аналіз і синтез мехатронних модулів.

Застосовуючи процедури функціонально-структурного й структурно-конструктивного аналізу, розробник оцінює ухвалені рішення, прагнучи добитися високого рівня синергетичної інтеграції елементів. Для використання методів автоматизованого проектування формують взаємозалежні функціональну (F-модель), структурну (S-модель) і конструктивну (C-модель) моделі мехатронного модуля. Визначення функції мехатронного модуля є проблемою концептуального проектування.

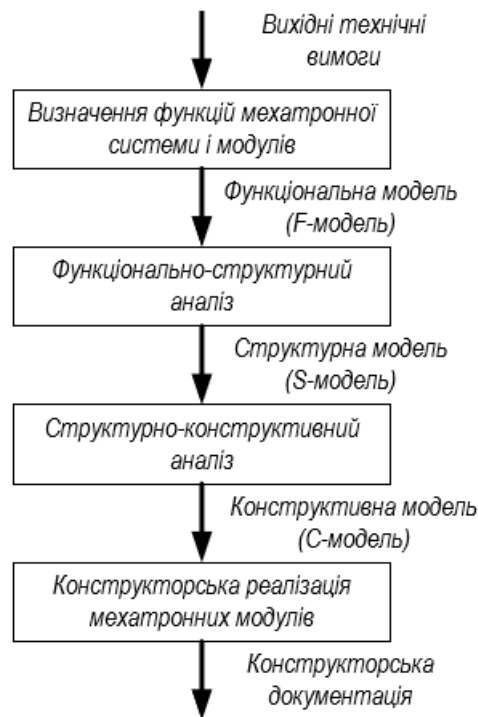


Рис. 3.11. Загальний алгоритм проєктування мехатронного модуля

Завдання з проєктування мехатронного модуля містить три основні етапи:

1) вибір варіантів структурних рішень модуля за його заданою функцією та їхній функціонально-структурний аналіз. Вхідною інформацією для цього етапу проєктування є F-модель, а на виході формується S-модель мехатронного модуля;

2) структурно-конструктивний аналіз конструкторських рішень і побудова C-моделі мехатронного модуля;

3) конструкторське втілення обраного варіанта модуля і розроблення конструкторської документації.

Завдання з проєктування мехатронного модуля полягає в знаходженні найкращої відповідності між заданою функцією й конструктивним виконанням. Процес проєктування мехатронної системи із мікропроцесорними пристроями з погляду створення її програмного й алгоритмічного забезпечення можна поділити на сім основних етапів (рис. 3.12).

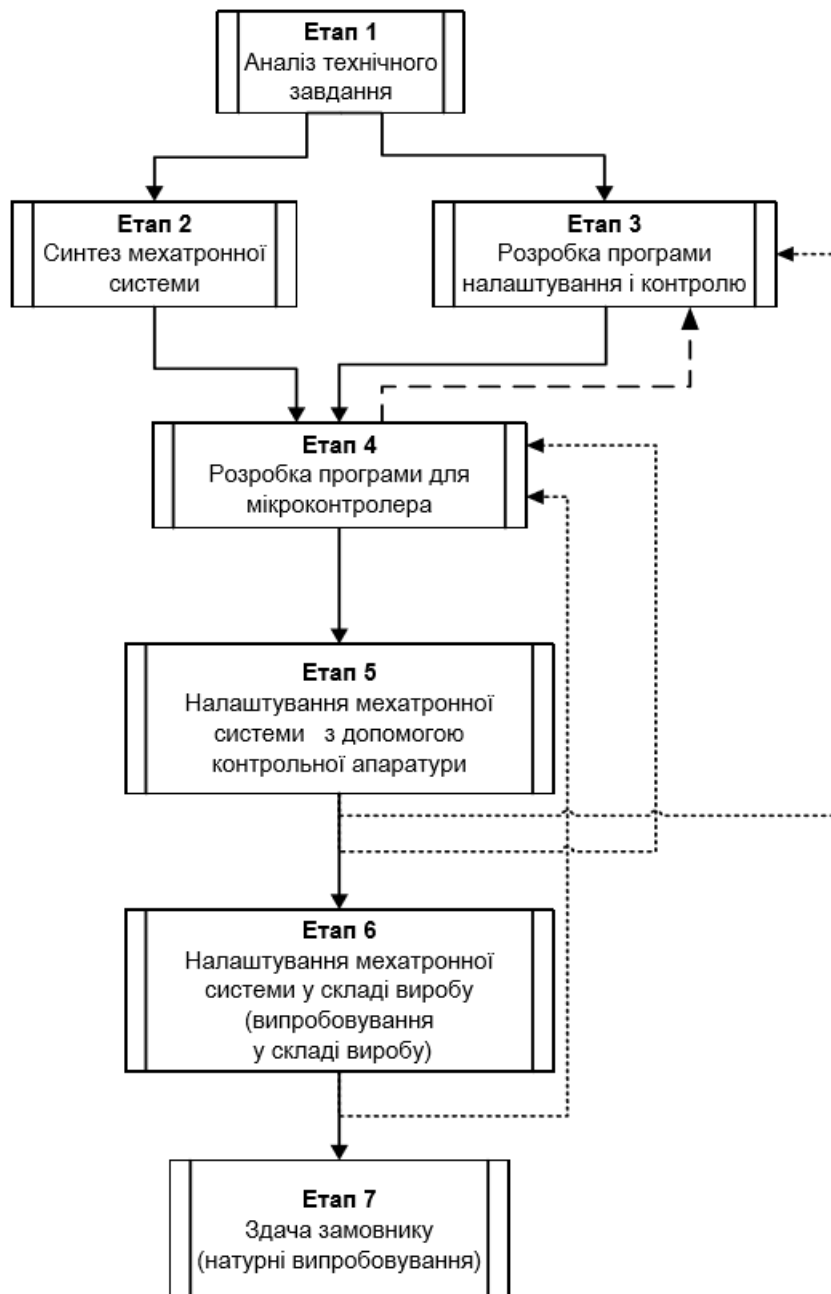


Рис. 3.12. Етапи проектування мехатронної системи мікропроцесорними пристроями

Етап 1. Аналіз технічного завдання

Зміст етапу:

- визначення структурної побудови мехатронної системи;
- визначення функціональної побудови мехатронної системи;
- вибір елементної бази системи, зокрема вибір мікроконтролера та його обв'язків;
- визначення переліку коригувальних пристроїв і законів керування, які планують реалізувати в мікроконтролері.

Етап 2. Синтез мехатронної системи (ескізний, технічний проєкти)

Етап синтезу (у широкому сенсі) можна представити у вигляді п'ятих кроків:

- енергетичний розрахунок силової частини мехатронної системи (остаточний вибір виконавчого пристрою, розглянутого на етапі аналізу технічного завдання);
- розроблення математичної моделі незмінної частини мехатронної системи (розроблення найпростішої математичної моделі для виконання «ручного» синтезу, нехтуючи всіма особливостями, внесеними мікропроцесорними пристроями керування);
- «ручний» синтез (синтез у вузькому сенсі) мехатронної системи (синтез лінійної системи традиційними «ручними» методами, наприклад, частотними, які бажано виконувати із застосуванням програмного забезпечення для автоматизації розрахунків і побудови «точних» частотних характеристик. Із усіх особливостей, внесених мікропроцесором у динаміку системи, тут доцільно брати до уваги тільки квантування за часом. Цей крок закінчується визначенням передаткових функцій коригувальних пристроїв і законів керування виконавчим пристроєм, які є вихідною точкою для подальшого моделювання);
- моделювання мехатронної системи (моделювання мехатронної системи з мікроконтролером за допомогою програмної моделі, яка реалізована на персональному комп'ютері);
- розроблення блок-схеми алгоритмічного забезпечення мехатронної системи (блок-схема являє собою графічне відображення програмно-апаратного втілення структурної побудови мехатронної системи у вигляді взаємодії джерел переривань і програм обробки цих переривань. Блок-схема слугує початковим матеріалом для розроблення керівної програми мікроконтролера).

Етап 3. Розроблення програми налаштування й контролю мехатронної системи (технічний, робочий проєкти)

Метою розроблення програми налаштування й контролю мехатронної системи є створення засобу, що дає змогу автономно (поза мехатронним виробом) через персональний комп'ютер налагоджувати й контролювати роботу мехатронної системи із мікроконтролером, причому без «перепрошивання» постійного запам'ятовувального

пристрою мікроконтролера у разі зміни параметрів коригувальних пристроїв, і за командами, що імітують роботу виробу.

Етап 4. Розроблення керівної програми мікроконтролера (робочий проєкт)

Метою розроблення керівної програми мікроконтролера є програмна реалізація коригувальних пристроїв і законів керування мікроконтролером. Очевидно, що між керівною програмою мікроконтролера і програмою налаштування й контролю повинна бути однозначна відповідність.

Етап 5. Налаштування мехатронної системи з мікроконтролером за допомогою контрольної апаратури (автономні випробування)

Метою етапу є попереднє налагодження програмно-апаратного забезпечення мехатронної системи. За результатами цього етапу можливе доопрацювання програмного забезпечення мехатронної системи.

Етап 6. Налаштування мехатронної системи з мікроконтролером у складі виробу (випробування у складі виробу).

Метою етапу є остаточне налагодження програмно-апаратного забезпечення мехатронної системи.

Етап 7. Здача замовникові (натурні випробування).

Метою етапу є здача мехатронної системи замовникові.

Контрольні запитання

1. Які основні принципи покладено в основу побудови мехатронних систем?
2. Які пристрої можуть бути складовою частиною машин з комп'ютерним керуванням рухом?
3. Які функції виконує пристрій комп'ютерного керування в мехатронній системі або модулі?
4. Поясніть суть мехатронного підходу до проєктування.
5. Які основні переваги мехатронного підходу до створення машин з комп'ютерним керуванням порівняно з традиційними способами?
6. Назвіть інтеграційні завдання, розв'язувані під час конструювання мехатронних пристроїв.
7. Опишіть особливості ієрархії рівнів інтеграції у мехатронних системах.

Лекція 4. Сучасні методи керування мехатронними модулями

Для керування мехатронними моделями застосовують різні системи автоматизованого управління (САУ).

Системи автоматизованого управління поділяють:

- за принципом керування: стежні, стабілізаційні та програмного керування;
- за структурою: розімкнені та зі зворотним зв'язком;
- за кількістю контурів керування: одноконтурні та багатоконтурні (з паралельними контурами, ієрархічні та з перехресними контурами);
- за властивістю керування в сталому режимі: статичні та астатичні;
- за принципом керування: із заданою настановою, з сигналом збурення та комбіновані;
- за фізичною реалізацією компонентів та сигналів: електронні, електромеханічні, гідравлічні, механічні, пневматичні, комбіновані, спеціалізовані (комп'ютеризовані).

Різні будови систем управління ґрунтуються на таких принципах автоматичного управління:

- відхиленням від установки або апертури;
- збурення (компенсуванням збурень);
- комбінованим регулюванням;
- адаптивним (самоналаштовувальним регулюванням).

Принцип керування та регулювання за відхиленням застосовується за двома критеріями.

Критерій 1. Мінімум середньоквадратичного відхилення (рис. 4.1).

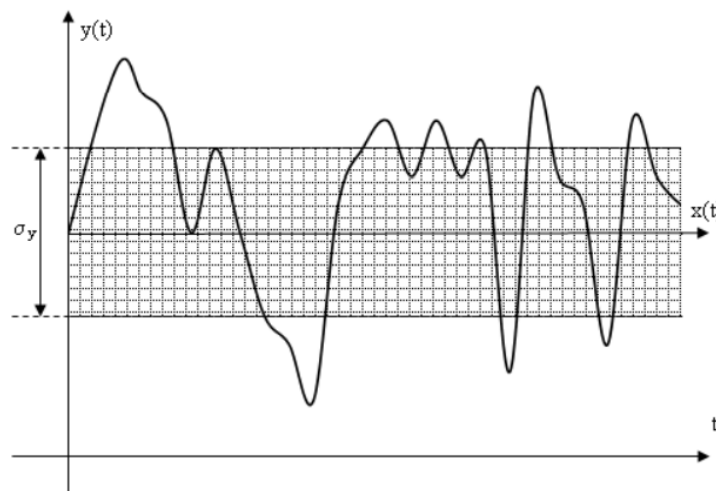


Рис. 4.1. Функція регулювання за принципом мінімум середньоквадратичного відхилення

Критерій 2. Обмеження порогового значення (рис. 4.2).

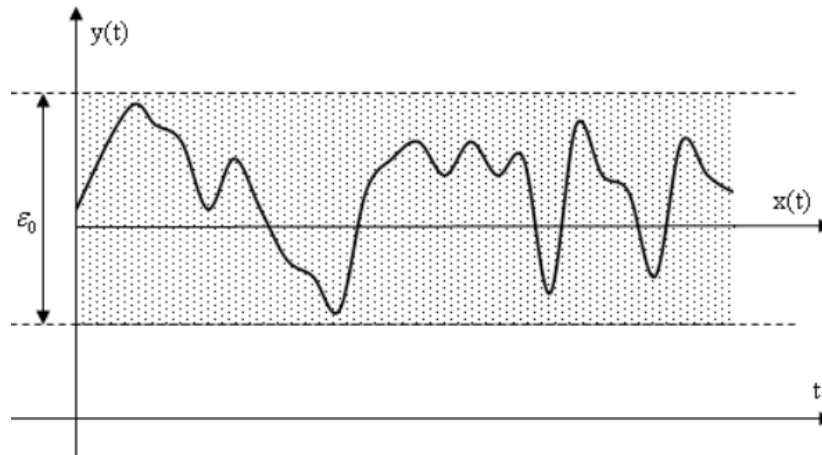


Рис. 4.2. Функція регулювання за принципом обмеження порогового значення

Критерій мінімуму середньоквадратичного відхилення (*Minimum Mean Square Error, MMSE*) полягає в тому, що для отримання оптимальної оцінки або прогнозу сигналу мінімізується очікуване значення квадрата відхилення між отриманою оцінкою і справжнім значенням сигналу:

$$\delta_y = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \bar{y})^2} \rightarrow \min, \quad (4.1)$$

де $\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$ – математичне сподівання (середнє значення) керованої функції.

Згідно з критерієм обмеження порогового значення встановлюють інтервальний критерій керування за відхиленням, що передбачає будь-яке значення вихідної функції об'єкта управління з будь-яким законом розподілу в заданих межах відхилення.

Принцип керування та регулювання за відхиленням реалізують замкнуті структури САУ із зворотним зв'язком (рис. 4.3).

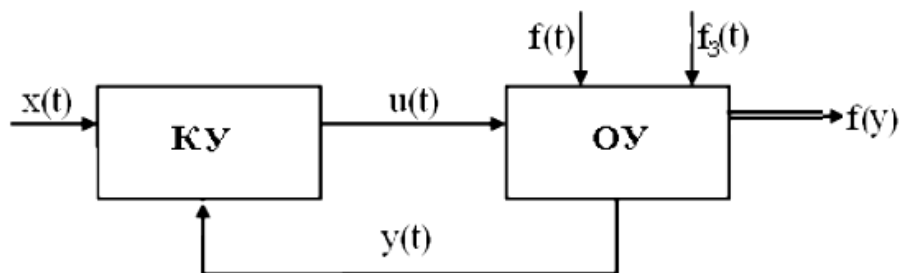


Рис. 4.3. Функціональна система САУ: ОУ – об'єкт управління; КУ – контролер

Структура САУ зі зворотним зв'язком (рис. 4.3) складається з двох функціонально пов'язаних компонентів: об'єкта управління (ОУ) та контролера управління (КУ), який виконує функції регулятора. Окрім цього на рис. 4.3 позначено: t – функція часу, $x(t)$ – функція управління; $u(t)$ – функція керування; $f(t)$ та $f(y)$ – вхідна та вихідна функції ОУ; $f_3(t)$ – функція вхідного випадкового збурення; $y(t)$ – інформаційна функція контролю відхилення стану ОУ, за якою відбувається регулювання.

Контролер автоматичного управління складається з АЦП (перетворювач аналогових сигналів у цифрові), спецпроцесора SP (виконує функції порівняння заданого стану $x(t)$ об'єкта управління з реальним станом $y(t)$ та реалізує закон регулювання), ЦАП (перетворює цифрові дані в аналогові), підсилювача ПС, інтерфейсного виконавчого механізму IBM (виконує управління об'єктом), сенсора S (перетворює регульований параметр ОУ в електричний сигнал) (рис. 4.4).

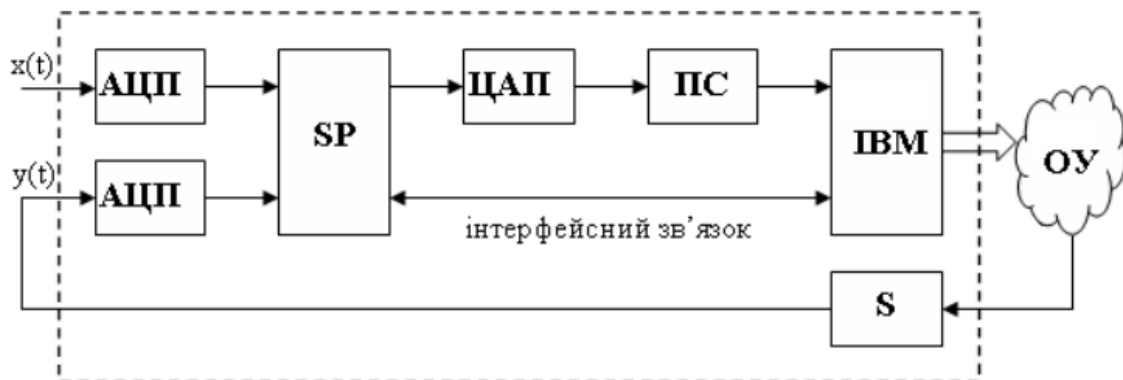


Рис. 4.4. Структура електронного контролера управління (автоматичного регулятора)

Принцип керування за збуренням полягає у тому, що керівна функція $u(t)$ формується залежно від вимірювання збурення $f_3(t)$.

САУ з компенсуванням збурення не мають зворотного зв'язку і поділяються на два класи:

- автоматичні системи компенсування збурення;
- автоматичні системи з програмним керуванням.

На рис. 4.5 зображено функціональну схему САУ, яка реалізує принцип керування за збуренням шляхом компенсування збурення.

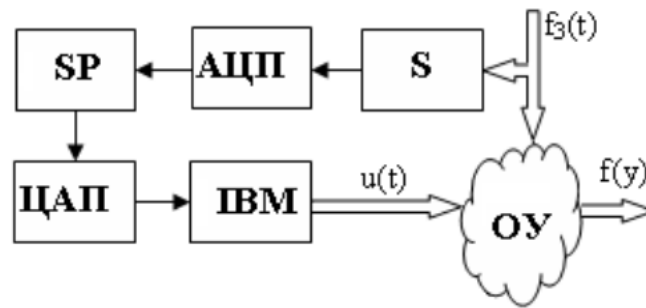


Рис. 4.5. Функціональна схема САУ за збуренням

Суть принципу керування за збуренням полягає у тому, що залежно від зміни збурення $f_3(t)$, контрольованого сенсором S на керований вхід ОУ подається керівна дія $u(t)$, що компенсує дію збурення. У цих САУ немає автоматичного контролю вихідної функції ОУ $f(y)$. Перевага таких САУ – відносна простота та підвищена надійність. Основними недоліками є складність динамічного контролю збурень і менша точність регулювання.

Функціональна схема САУ з програмним керуванням показана на рис.4.6.

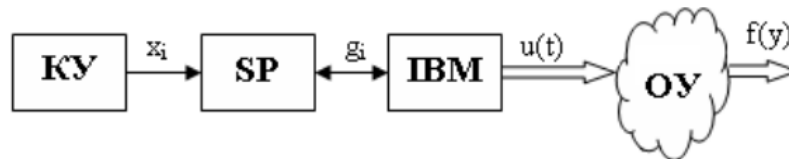


Рис. 4.6. Функціональна схема САУ з програмним керуванням
(g_i – інтерфейсний зв'язок)

САУ з програмним керуванням мають розімкнену структуру і передбачають жорстке виконання ОУ певної послідовності команд та змін режимів роботи. Такими об'єктами є об'єкти з квазістаціонарними станами.

САУ за принципом комбінованого регулювання мають структуру як на рис. 4.7.

Особливістю такої структури є наявність двох сенсорів, які контролюють характер збурень та зміну станів ОУ. Сигнали сенсорів опрацьовуються спеціальним процесором, відповідно до заданої програми $x(t)$. Перевагою таких систем є підвищення стійкості процесів регулювання, а недоліком – ускладнення алгоритмів роботи спецпроцесора.

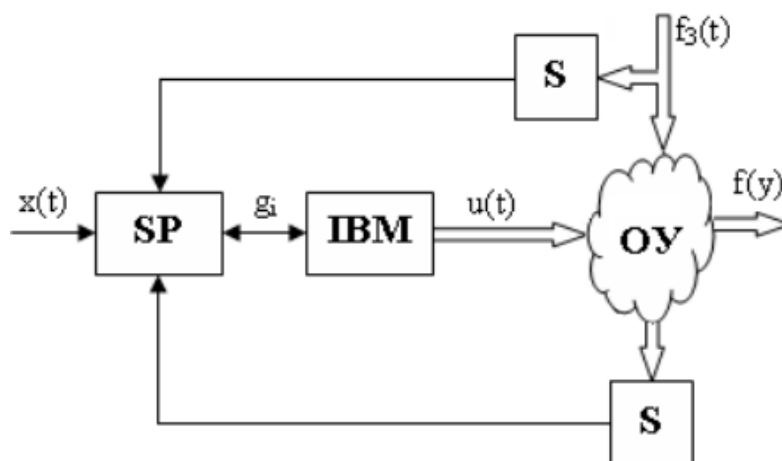


Рис. 4.7. Структура САУ з комбінованими управлінням ОУ

Принцип з адаптивним самоналаштуванням регулювання вирішує складніші задачі, а саме підтримку екстремуму керованої величини або підтримку оптимальної роботи системи регулювання за критерієм максимальної швидкодії. До САУ на адаптивних принципах керування належать: екстремальні системи; системи з налаштуванням керівних ланок; самооптимізувальні. Функціональну схему САУ з самоналаштуванням керівних ланок наведено на рис. 4.8. Особливістю такої САУ є наявність додаткового регулятора (DR), який після опрацювання даних $x(t)$, $f_3(t)$ та $y(t)$ виконує оперативну корекцію (самоналаштування) та адаптацію характеристик системи регулювання за програмованою функцією установок $g(t)$. Прикладом такої системи є автопілот літака. Недоліком таких систем є висока складність, а також менша динамічність внаслідок виконання обчислень складних алгоритмів.

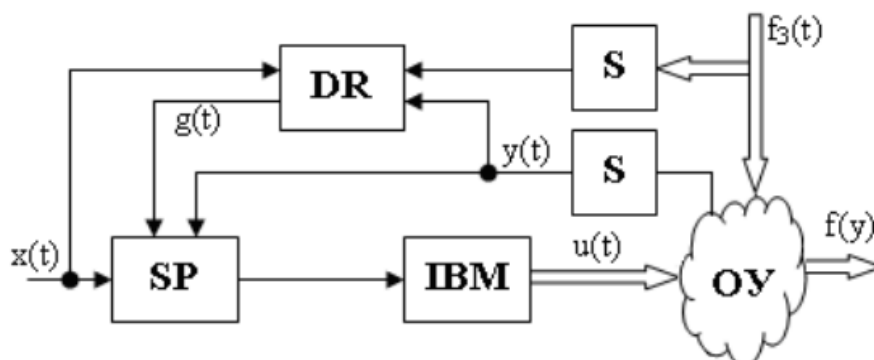


Рис 4.8. Функціональна схема САУ з самоналаштуванням керівних ланок

У керуванні мехатронними системами велику увагу приділяють процесам формування різних видів сигналів, які використовують як для управління, так і для діагностування параметрів стану механічної системи.

Сигнали можуть бути аналоговими та цифровими.

Аналоговий сигнал – це сигнал неперервний на всьому проміжку часу. Цифровий сигнал – дискретний сигнал з певним значенням інформативного параметра у цифровій формі. Найпростішим типом аналогового сигналу є рівень напруги з прямою (хоча і не обов'язково лінійною) кореляцією до вхідного стану. Другий тип – це сигнал із широтно-імпульсною модуляцією (ШІМ). Третій тип – це хвильовий сигнал (рис. 4.9). Цей тип сигналу може змінюватися (модулюватися) або за амплітудою (рис. 4.9, б), або за частотою (рис. 4.9, в), а в деяких випадках за обома параметрами, що відображають зміни в стані, який контролюють.

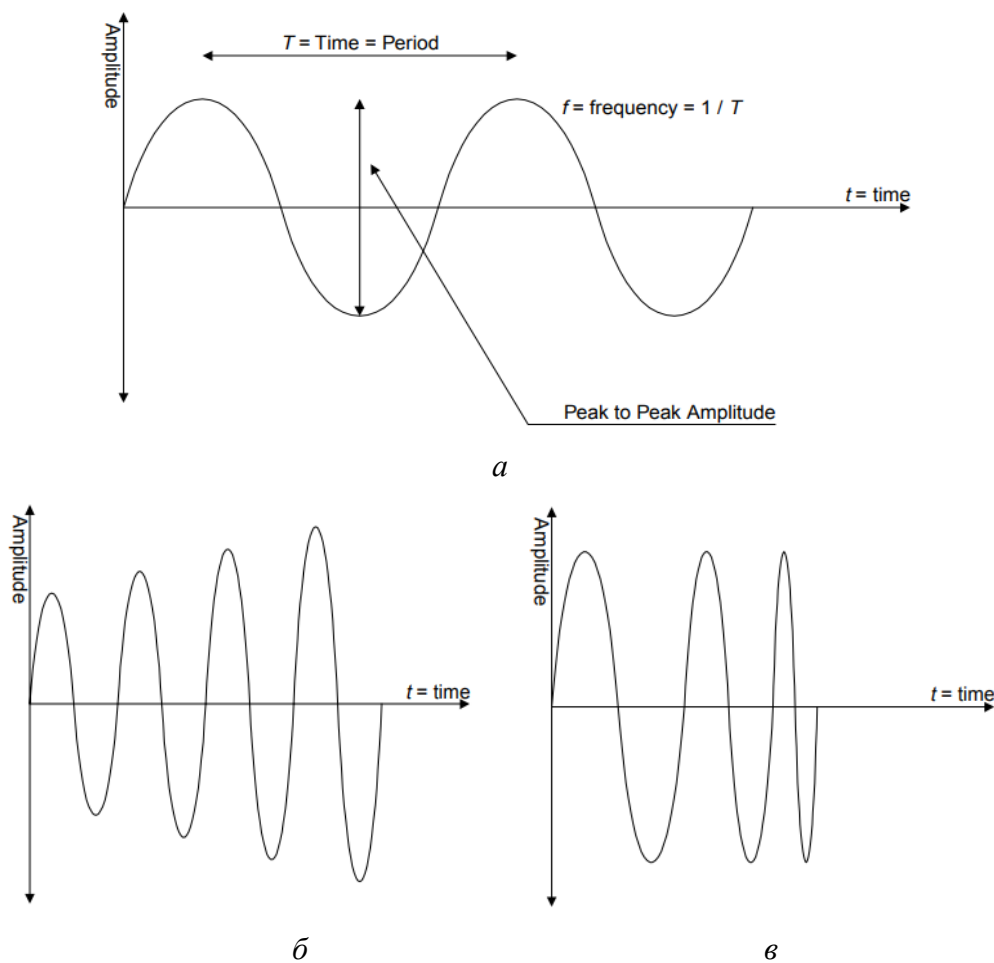


Рис. 4.9. Види аналогових хвильових сигналів: а – безперервний синусоподібний сигнал; б – модульований за амплітудою; в – модульований за частотою

Цифрові сигнали є зображенням дискретного сигналу, який часто видобувається шляхом квантування аналогового сигналу (рис. 4.10). В комп'ютерах та інших цифрових системах, цифровий сигнал є хвилею, що перемикається між двома рівнями напруги (0 та 1). У більшості

комп'ютерних програм цифровий сигнал зображується у вигляді двійкових чисел і тому точність квантизації вимірюється у бітах. Наприклад, 4-бітова система забезпечить підтримку $2^4 = 16$ дискретних значень, 7-бітова – $2^7 = 128$, 16-бітова – $2^{16} = 65536$ дискретних значень і т. д.

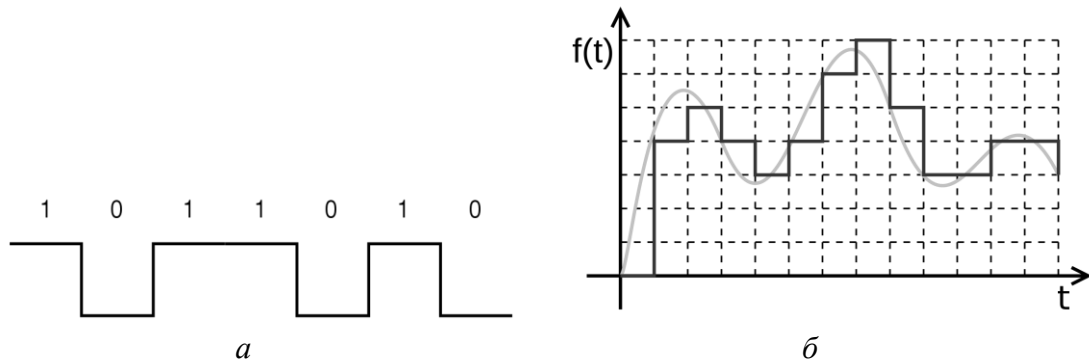


Рис. 4.10. Види цифрових сигналів: *a* – двійковий; *б* – квантований

В електричному сигналі напруга, струм або частота сигналу можуть змінюватися для подання інформації.

Широтно-імпульсна модуляція (ШІМ – англ. pulse-width modulation, PWM), або модуляція за тривалістю імпульсів, – процес керування шириною (тривалістю) височастотних імпульсів за законом, який задає низькочастотний сигнал. В електроніці це може бути керування середнім значенням вихідної напруги шляхом зміни тривалості замкнутого стану електронного (електромеханічного) ключа, наприклад, у схемі ключового стабілізатора напруги.

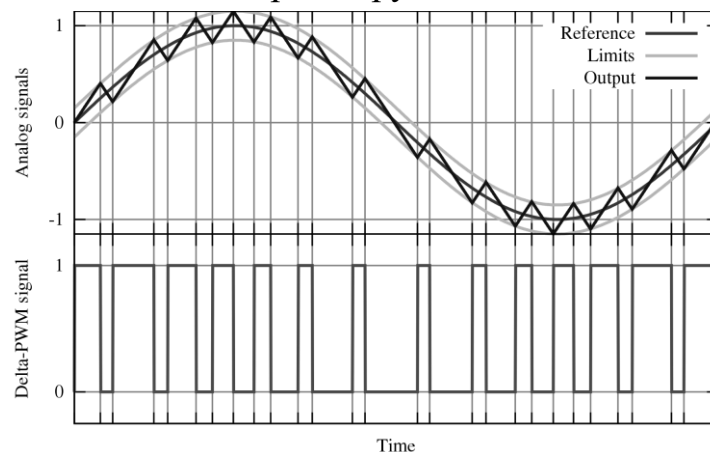


Рис. 4.11. Моделювання в ШІМ вихідного синусоїдального сигналу

Розглянемо завдання управління мехатронними системами, які використовуються у виробничих машинах та комплексах і виконують основні технологічні функції. До таких систем управління ставлять дуже жорсткі вимоги, оскільки режими управління визначають перебіг технологічного процесу, а отже, впливають на якість виробу. В

мехатроніці ставиться завдання з управління координованими функціональними (механічними) рухами машин.

Розглянемо функціональну схему пристрою з комп'ютерним керуванням (рис. 4.12).

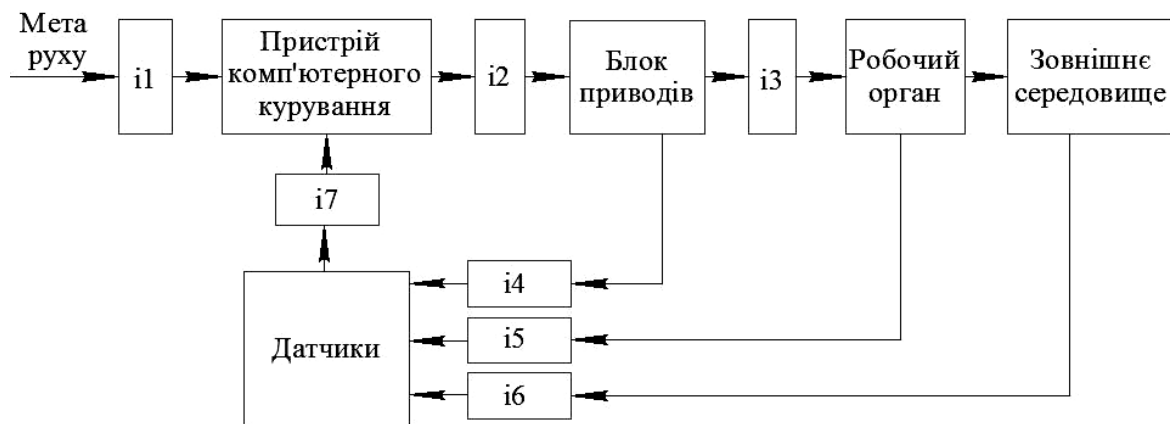


Рис. 4.12. Функціональна схема пристрою з комп'ютерним керуванням

Завдання з керування полягає у виконанні бажаного руху робочого органу, який цілеспрямовано впливає об'єкт керування. При цьому з боку зовнішнього середовища об'єкт відчуває збурювальний вплив. У загальному випадку об'єктом управління в мехатроніці є складна багатозв'язкова система з інтерфейсами i1-i7, до складу якої входить блок виконавчих приводів, механічний пристрій з робочим органом, блок сенсорів (датчиків).

Наведена структура об'єкта керування визначає вимоги та постановку завдання з управління. Відтворення заданих рухів мехатронними модулями ґрунтується на дотриманні класичних вимог теорії автоматичного керування: стійкості, точності та якості процесу керування. Крім того, необхідно додатково зважати на такі специфічні особливості мехатронних систем:

- рух робочого органа зазвичай досягається взаємопов'язаними переміщеннями декількох виконавчих приводів і ланок механічного пристрою;
- завдання з керування мехатронною системою має бути вирішене у просторі (знайдені оптимізовані траєкторії руху всіх ланок, зокрема робочого органа) та у часі (визначено та реалізовано бажані швидкості, прискорення та зусилля для всіх приводів системи);
- параметри збурювальних впливів, доданих до робочого органа та окремих мехатронних модулів, для багатьох технологічних завдань заздалегідь не визначені;

– складність побудови адекватних математичних моделей мехатронних систем (особливо багатозв'язних, що містять динамічну модель технологічного процесу).

Розмірність задач з управління мехатроною системою визначають кількістю незалежно керованих приводів системи. Для планування заданого руху мехатронної системи треба розв'язати обернену задачу про положення механізму. Суть цього завдання полягає у визначенні потрібних переміщень ланок системи за заданим рухом робочого органа.

Багатозв'язність системи означає, що рух кожної ланки впливає на рух інших ланок. Взаємовплив відбувається через механічний пристрій, через загальне джерело енергії, а також внаслідок природних та штучних динамічних зв'язків між каналами керування в блоці приводів. Отже, слід формувати керування рухом мехатронної системи (особливо на високих швидкостях, коли вплив динамічних факторів є суттєвим) з огляду на взаємозв'язки між ланками. Проблемним є питання організації зворотних зв'язків під час управління багатоланковими мехатронними системами. Технічно найпростіше встановлювати датчики положення та швидкості в приводних модулях, однак потім потрібно обчислити в реальному часі фактичне переміщення робочого органа. Причому цей комп'ютерний розрахунок потребує побудови адекватної динамічної моделі системи відповідно до дуже складних для аналітичної оцінки факторів:

- всіх дійових сил (керівних моментів приводів, сил тертя, зовнішніх сил і моментів, відцентрових і коріолісних сил);
- первинних похибок системи (пружних деформацій ланок, люфтів у механічних передачах, похибок виготовлення та складання, вузлів), що визначають її інтегральні характеристики точності;
- змінних параметрів об'єкта управління (наведених моментів інерції та мас механізму та навантаження).

Характерною особливістю мехатронних систем для автоматизованого машинобудування є можливість поділу завдань програмного управління рухом на просторову та часову. Це означає, що траєкторію переміщення робочого органа в просторі і його контурну швидкість можна планувати окремо, застосовуючи різні критерії оптимізації. Оптимізацію закону руху технологічної машини в часі зазвичай виконують за критеріями продуктивності, точності обробки, а також за економічними та комбінованими показниками.

Розглянемо процес керування безколекторним двигуном із датчиком Голла. Для управління двигуном застосуємо мостову схему із замикальними контактами (рис. 4.13).

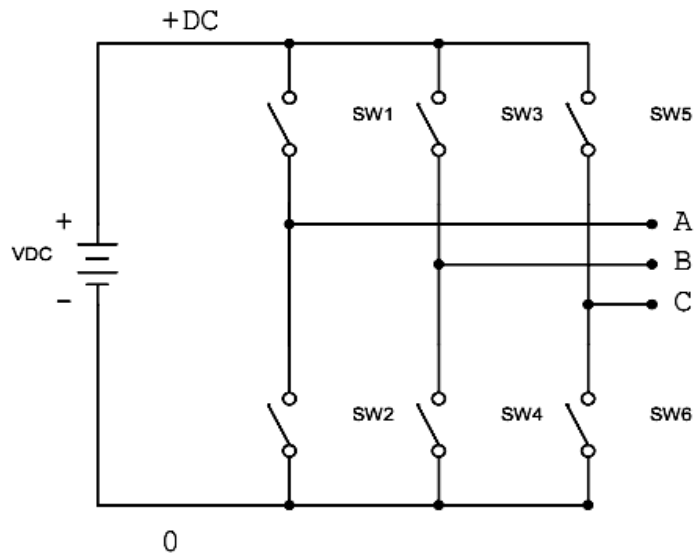


Рис. 4.13. Мостова контактна схема керування безколекторним електродвигуном

Безколекторний двигун має три входи (три фази), на які потрібно подавати в різні моменти часу живлення («+» або «-»). Цього можна досягти за допомогою електронних ключів, ввімкнених за мостовою схемою: **SW1** вмикає «+» на фазу **A**, а замикаючи **SW6**, вмикаємо «-» на фазу **C**, що дасть проходження струму від джерела живлення до фази **A** та **C**. Для зворотного напрямку вмикаємо **SW5** та **SW2**. В такому разі струм проходитиме від джерела живлення через фази **C** та **A** у зворотному напрямку. Під час роботи двигуна одночасно повинен бути відкритий лише один верхній ключ та один нижній ключ. Для зміни стану потрібно відразу вимкнути пару ключів, дочекатися закриття ключів і лише після цього вмикати іншу пару ключів (табл. 4.1).

Після ввімкнення ключів, як про це йшлося раніше, на двигун подається повна напруга живлення. При цьому двигун розвиває максимальні оберти (потужність). Щоб забезпечити керування двигуном, потрібно регулювати напругу живлення двигуна. Зміна поточної напруги відбувається за допомогою широтно-імпульсної модуляції (ШІМ). Для цього перемикають контакти з фіксованою змінною частотою. Таким чином, змінюється напруга від мінімального до максимального значення напруги живлення (рис. 4.14). Щоби домогтися керування обертами двигуна, потрібно накласти сигнал ШІМ на сигнали, які надходять на ключі.

Таблиця 4.1

Послідовність ввімкнення ключів залежно від сигналів датчиків Голла

Hall Sensors Value (Hall_CBA)	Phase	Switches
101	A-B	SW1 ; SW4
001	A-C	SW1 ; SW6
011	B-C	SW3 ; SW6
010	B-A	SW3 ; SW2
110	C-A	SW5 ; SW2
100	C-B	SW5 ; SW4

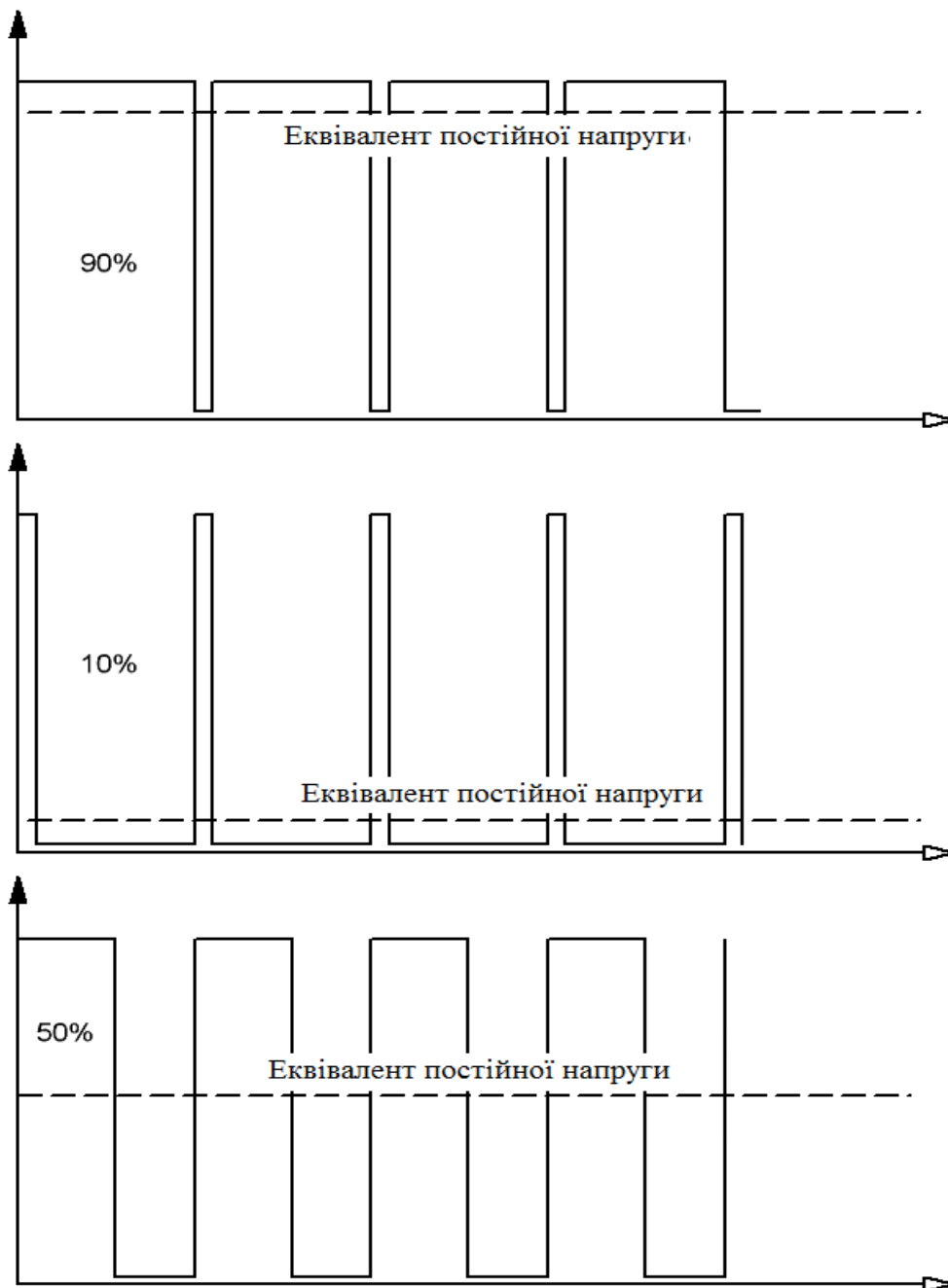


Рис. 4.14. ШІМ-регулювання

У випадку з двигунами без датчиків вибір частоти ШІМ-сигналу відіграє важливу роль. Деякі мікроконтролери можуть апаратно формувати ШІМ-сигнал на кількох своїх виходах, що дає змогу застосовувати схеми мікроконтролерів для логічних елементів. Частоту ШІМ-сигналу зазвичай обирають від 4 до 80 кГц. Під час ввімкнення та вимкнення ключів відбуваються перехідні процеси, внаслідок чого на ключах виділяється додаткове тепло. Що вища частота ШІМ-сигналу, то більше перехідних процесів за одиницю часу і більше втрат на ключах. Занадто мала частота може бути неефективною або не забезпечувати достатньої плавності регулювання.

Для зменшення втрат внаслідок перехідних процесів на ключах ШІМ-сигнал можна подавати тільки на нижні або тільки на верхні ключі. Прямі втрати на ключах (крім перехідних процесів) можна розрахувати за такою формулою:

$$Q=RI^2, \quad (4.2)$$

де Q – кількість виділеного тепла; R – опір відкритого ключа; I – струм, що протікає через ключ.

Очевидно, що чим менше опір ключів, тим менші втрати на ключах, а зменшення опору ключів сприятиме підвищенню загального ККД та зменшенню тепловиділення на ключах. Рівень втрат на ключах має квадратичну залежність від струму. Зменшити струм, зберігши при цьому загальну потужність, можна, якщо підвищити напругу живлення двигуна. Як приклад розглянемо два варіанти:

1. Живлення: 50В, струм: 100А, опір ключів: 0,001 Ом, втрати на ключах $= 0,001 \cdot 100^2 = 10$ Вт;
2. Живлення: 100В, струм: 50А, опір ключів: 0,001 Ом, втрати на ключах $= 0,001 \cdot 50^2 = 2,5$ Вт.

Час ввімкнення ШІМ також залежить від конструкції електродвигуна, що пов'язане з фізикою перебігу процесів, тобто електродвигун з усіма передачами повинен встигнути виконати задану роботу, яка визначена системою керування. Після подачі напруги на обмотки двигуна струм в обмотках не може миттєво збільшитися, оскільки обмоткам електродвигуна властива індуктивність. Від моменту подачі напруги до досягнення струмом максимального значення повинен минути деякий час. Аналогічно після зняття напруги знадобиться деякий час, поки струм зменшуватиметься до нуля. Цей час

залежить від індуктивності обмотки двигуна та інших конструктивних особливостей. Таким чином, дія обмотки двигуна дещо запізнюється за сигналом керування. Щоб компенсувати цю затримку, керівний сигнал на ключі подають з випередженням. Випередження сигналу керування виражають у куті випередження. Кут випередження може бути від 0 до 30 градусів. Кут випередження може відрізнятися залежно від моделі двигуна. Точність встановлення кута випередження значно впливає на роботу високооборотних двигунів. На малих швидкостях точність установлення кута випередження є менш критичною. Налаштування кута випередження (timing) виконується або переміщенням датчиків, або коригується програмно засобами регулятора. Якщо двигун має реверсивний режим, рекомендують застосовувати програмний метод.

Щодо безколекторних двигунів без датчиків положення або енкодерів, то в них визначення положення ротора відбувається шляхом вимірювання ЕРС на вимкненій фазі. В кожний момент часу під'єднано одну з фаз, тоді інша залишається не ввімкненою і під час обертання ротора в двигуні на цій фазі створюється ЕРС, яка змінюється. Вимірюючи напругу на цій фазі, можна визначити момент перемикавання до наступного положення ротора. Зазвичай визначають момент переходу напруги на вимкненій фазі через нульову точку (половину напруги живлення), тобто коли напруга на фазі зрівняється із середньою точкою. Цей метод добре працює за досить високих обертів двигуна, а за низьких обертів ЕРС може виявитися недостатньою для чіткого визначення положення ротора (рис. 4.15).

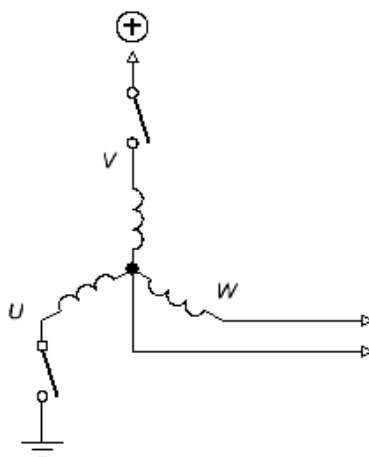


Рис. 4.15. Схема двигуна BLDC з трьома фазами

Схема керування із середньою точкою означає, що коли ключі відкриті, то через фази протікає струм, а середня точка «прив'язана» до

напруги живлення. У цей момент на вимкненій фазі двигуна виконуються вимірювання. Очевидно, що за розімкнених ключів, вимірювання напруги на вимкненій фазі не дадуть результату. Найчастіше середня точка є недоступною, тому немає можливості фізично під'єднатися до неї без розбирання двигуна. Уникнути цієї ситуації допоможе створення віртуальної середньої точки (рис. 4.16).

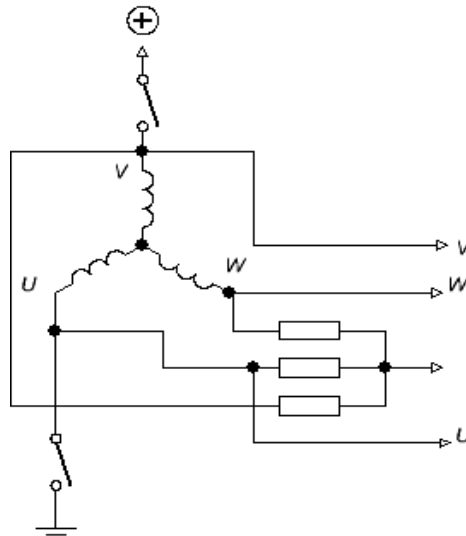


Рис. 4.16. Схема ввімкнення двигуна BLDC з віртуальною середньою точкою

Ця схема дуже проста і застосовується дуже часто, але має певні недоліки. Через ШІМ-керування напруга в середній точці не завжди постійна і коливається у широкому діапазоні. Для того щоб мікроконтролер зміг виміряти напругу, застосовують ланцюги узгодження сигналів – ділянки напруги та RC-фільтри для згладжування коливань (рис. 4.17).

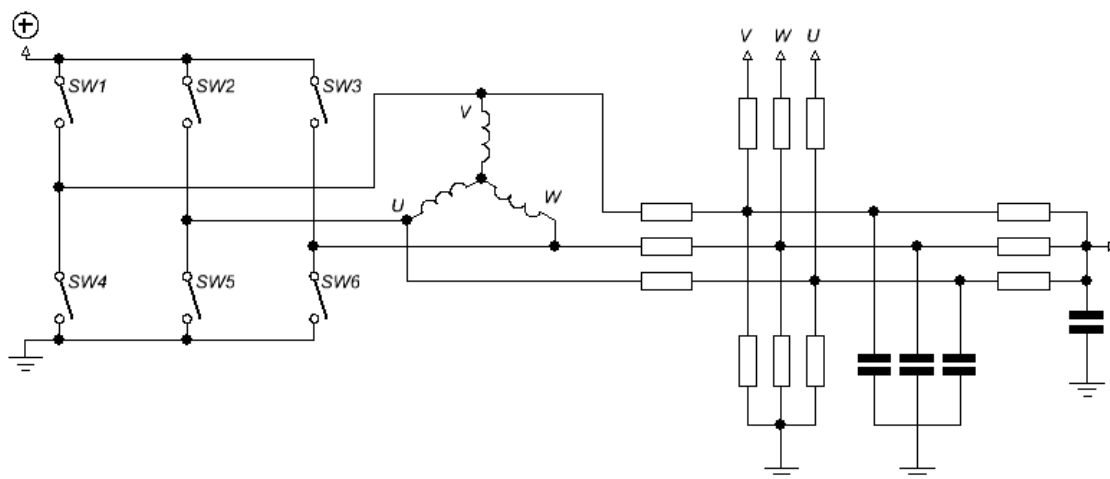


Рис. 4.17. Схема ввімкнення двигуна BLDC з RC-фільтрами

Якщо BLDC-двигун живиться напругою 24 вольти, тоді в середній точці напруга може бути в межах від 0 до 24 В. Мікроконтролери

зазвичай мають межу виміру 5 В, тому застосовують схему узгодження рівнів сигналів і, додатково, схеми захисту. Наявність ділника напруги призводить до зниження чутливості двигуна на малих обертах, а наявність фільтрів – до затримки. Затримка є причиною похибок у визначенні положення ротора і може стати критичною для керування двигунами на високих швидкостях. Інша схема живлення BLDC-двигуна застосована на стабілізаторі напруги середньої точки, що встановлюється окремо і залежить від напруги живлення (рис. 4.18).

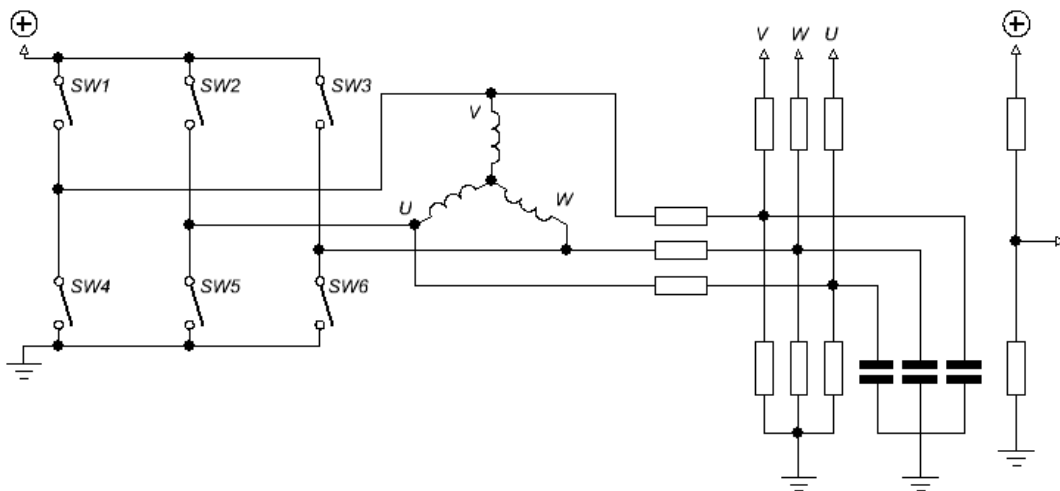


Рис. 4.18. Схема ввімкнення двигуна BLDC зі стабілізатором напруги

Алгоритм комутацій фаз двигуна BLDC такий. На вимкненій фазі за допомогою аналого-цифрового перетворювача (АЦП) вимірюють напругу і порівнюють із середньою точкою. Оскільки АЦП виконує перетворення сигналу протягом деякого часу, це призводить до похибки обчислення. Можна також використовувати компаратори, адже не обов'язково знати значення напруги, а важливо визначити сам момент переходу через середню точку. Компаратор спрацьовує саме в момент переходу напруги через середню точку і може генерувати переривання мікроконтролера. Розглянемо часову діаграму на рис. 4.19.

На першому етапі Р1 на фазу V подається «плюс» і «мінус» – на фазу U. При цьому напруга на фазі W починає зростати й у період Р1 перетинає нульову точку – половину напруги між фазами V і U, тобто половину напруги живлення. Як видно з діаграми, зміну стану ключів потрібно виконати на половині періоду між подією перетину нульової точки (ZC).

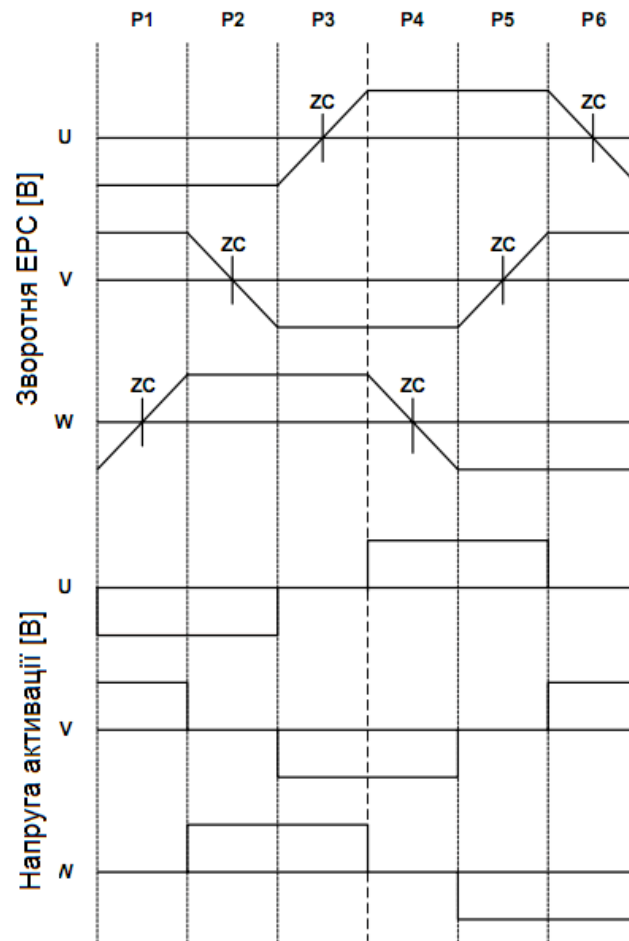


Рис. 4.19. Часова діаграма перемикання фаз BLDC-двигуна

Після перемикання стану ключів (етап P2) вимірювання виконуємо на вільній фазі V. При цьому напруга на вільній фазі може зростати або спадати. Це доведеться мати на увазі під час роботи компараторів. Таким чином, регулятор повинен пам'ятати, на якому кроці від P1 до P6 знаходиться, і переходити до наступного стану відповідно до розрахунків часу, обчислених між подіями ZC. В табл. 4.2 відповідності стану ключів та вимірюваних фаз для кожного з шістьох положень ротора.

Таблиця 4.2

Таблиця комутації фаз

Етап	Живлення +	Живлення –	Відкриті ключі	Вимірювання на фазі
P1	V	U	SW1, SW5	W↑
P2	W	U	SW3, SW5	V↓
P3	W	V	SW3, SW4	U↑
P4	U	V	SW2, SW4	W↓
P5	U	W	SW2, SW6	V↑
P6	V	W	SW1, SW6	U↓

Найчастіше у приводах для визначення положення ротора вимірювання струму і напруги синхронізують із сигналом ШІМ. Припустімо ситуацію, коли частота ШІМ 8 кГц, шпаруватість – 100% (повний газ), двигун обертається зі швидкістю, за якої частота комутацій теж дорівнює 8 кГц. Тобто від моменту перемикання ключів до наступного перемикання минає рівно один період ШІМ-сигналу. Якщо швидкість обертання зросте і частота комутацій перевищить частоту ШІМ, виникне ситуація, коли ШІМ-сигнал буде утримувати ключі відкритими довше, ніж потрібно, і протидіятиме обертанню двигуна. Крім того, вимірювання напруги на вільній фазі синхронізовані частотою ШІМ-сигналу, тому немає технічної можливості обчислити швидкість комутації, вищої за частоту ШІМ. Тобто контролер теоретично не зможе керувати двигуном, якщо його вал обертається зі швидкістю, за якої частота комутацій перевищує частоту ШІМ-сигналу. Таким чином, бажано, щоби частота ШІМ-сигналу у кілька разів перевищувала частоту комутацій. Наприклад, маємо регулятор із частотою ШІМ f кГц і двигун із N магнітами. Максимальна теоретично можлива частота обертання вала двигуна дорівнюватиме:

$$V=(f/6/(N/2)*60), \quad (4.3)$$

де f – частота ШІМ у герцах; 6 – кількість комутацій за один електричний оберт; N – кількість магнітів; 60 – кількість секунд за хвилину.

Таким чином, якщо потрібно керувати багатополюсним двигуном на високих обертах, доведеться використовувати регулятор із вищою частотою ШІМ. Наприклад, щоби розкрутити двигун з 24 магнітами до 10 000 об/хв, знадобиться регулятор із частотою ШІМ, не нижчою за 24 кГц. Потрібно також пам'ятати: що вища частота ШІМ, то більше перехідних процесів відбувається на ключах за одиницю часу. Це може призвести до збільшення втрат і збільшення тепловиділення на ключах.

В асинхронних двигунах змінного струму для їхнього кращого використання та отримання високих енергетичних показників роботи (коефіцієнтів потужності, корисної дії та перевантажувальної здатності) одночасно зі зміною частоти треба регулювати і напругу, яка підводиться до двигуна. Співвідношення частоти і напруги дістало назву закону частотного керування. Обираючи співвідношення між частотою і напругою, яка підводиться до статора асинхронного двигуна, найчастіше

виходять з умови збереження перевантажувальної здатності асинхронного двигуна, тобто кратності критичного моменту M_k до моменту статичного навантаження M_c , для будь-якої з регулювальних механічних характеристик:

$$\lambda = \frac{M_k}{M_c} = \text{const}. \quad (4.4)$$

Нехтуючи падінням напруги на обмотці статора і зробивши деякі припущення, отримаємо основний закон зміни напруги за частотного регулювання швидкості асинхронного двигуна:

$$\frac{U_{1j}}{U_{1k}} = \frac{f_{1j}}{f_{1k}} \sqrt{\frac{M_c(\omega_j)}{M_c(\omega_k)}}, \quad (4.5)$$

де U_{1j} – фазна напруга джерела живлення за частоти f_{1j} ; $M_c(\omega_j)$ – момент статичного навантаження на валу двигуна за швидкості $\omega_f \approx 2\pi j_{1j}/p$.

Взявши один із режимів роботи двигуна за номінальний, тобто вважаючи, наприклад, що за $f_{1k} = f_{1н}$ до затискачів обмотки статора прикладено номінальну напругу $U_{1н}$ і при цьому двигун розвиває номінальний момент, основний закон зміни напруги за частотного регулювання можна записати у вигляді виразу

$$\frac{U_{1j}}{U_{1н}} = \frac{f_{1j}}{f_{1п}} \sqrt{\frac{M_c(\omega_j)}{M_н}}, \quad (4.6)$$

або у відносних одиницях:

$$U_{1*} = f_{1*} \sqrt{M_{c*}}, \quad (4.7)$$

де $U_{1*} = U_1/U_{1н}$; $f_{1*} = f_1/f_{1н}$; $M_{c*} = M_c/M_н$; U_1 , M_c – значення напруги на статорі й статичного моменту, відповідні значенню регульованої частоти f_1 .

З отриманих виразів випливає, що закон зміни напруги U_1 визначається не тільки частотою джерела живлення f_1 , а й характером зміни моменту статичного навантаження на валу двигуна внаслідок зміни кутової швидкості.

Найчастіше трапляються такі закони частотного керування:

– якщо $M=\text{const}$ $U_{1*} = f_{1*}$ або $U_1/f_1 = \text{const}$, тобто за постійного значення моменту статичного навантаження напруга живлення повинна змінюватися пропорційно її частоті;

– якщо $P_c = M_c \omega = \text{const}$ $U_{1*} = \sqrt{f_{1*}}$ або $U_1/\sqrt{f_1} = \text{const}$, тобто за постійного значення потужності статичного навантаження напруга

джерела живлення повинна змінюватися пропорційно квадратному кореню від значення частоти;

– за вентиляторного навантаження $U_{1*} = f_{1*}^2$ або $U_1 / f_{1*}^2 = \text{const}$, тобто за вентиляторного навантаження напруга джерела живлення повинна змінюватися пропорційно квадрату значення частоти.

У разі зміни частоти живлення, окрім зміни швидкості двигуна, відбувається зміна і струму намагнічення. Підтримання постійного струму намагнічення є можливим завдяки налаштуванню бусту, який дасть змогу підвищити вхідну напругу двигуна до потрібної величини. Потрібне значення напруги залежить від відношення опору статора R_1 до реактивного опору X_n , частоти f_1 інвертора вихідної напруги, завантаження двигуна. Для того щоби правильно встановити буст, треба виконати таке:

– виміряти завантаження інвертора у режимі неробочого ходу за номінальної частоти обертання;

– установити буст так, щоб завантаження інвертора було однаковим як за низьких частот (5...10 Гц) і роботи без навантаження, так і за номінальної кутової швидкості.

Таким чином, струм намагнічення залишається майже постійним на всіх частотах у режимі неробочого ходу. Щоб уникнути труднощів, які виникають у процесі установлення бусту, усі серії нових інверторів обладнані автоматичним установленням бусту (автобуст). Потрібна величина бусту автоматично визначається за даними двигуна і вимірювання струму двигуна. За потреби цю функцію можливо вимкнути.

Застосовують два способи встановлення бусту:

- установлення постійного бусту (рис. 4.20, а);
- установлення обмеженого в часі бусту (дельта-бусту) (рис. 4.20, б).

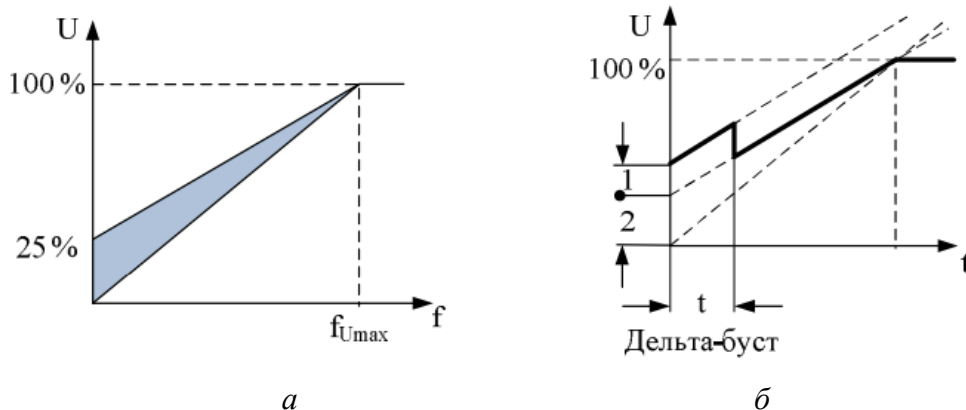


Рис. 4.20. Налаштування постійного бусту (а) і комбінації бусту і дельта-бусту (б): 1 – дельта-буст; 2 – буст

Дельта-буст можна використовувати для запобігання збільшенню початкового (пускового) моменту, оскільки він активний тільки у визначений інтервал часу від початку пуску. Оскільки двигун працює з постійним струмом, установлений буст і встановлена величина частоти 0 Гц, тоді за деяких умов це може призвести до перегрівання двигуна. У разі використання комбінації бусту і дельта-бусту (рис. 4.20, б) виконують додавання величини.

Перетворювачі частоти дають змогу змінювати швидкість обертання електродвигуна за постійного моменту (рис. 4.21). Для швидкої зупинки або зниження швидкості механізму, якому надає руху електродвигун, крім механічних використовують й електричні способи гальмування.

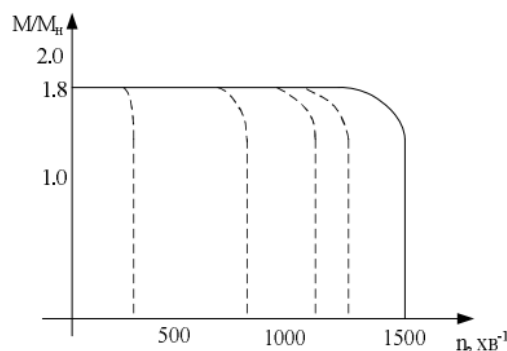


Рис. 4.21. Механічна характеристика електродвигуна в разі використання перетворювача частоти

Суть електричних способів гальмування полягає в тому, що електрична машина в цей період з двигунного режиму переводиться в генераторний і створює електромагнітний момент, спрямований проти руху.

Найпростіший спосіб зупинки електродвигуна – від'єднання від мережі живлення і зупинення за інерцією. При цьому час до повної зупинки не регулюється, він залежить від інерційних властивостей двигуна і його навантаження.

Регульованого гальмування досягають генераторним способом, який полягає в тому, що перетворювач з потрібною швидкістю зменшує вихідну частоту до заданого значення. При цьому двигун працює в генераторному режимі, перетворюючи кінетичну енергію обертання на електричну. Тобто перевагою генераторного гальмування є можливість, задавши потрібний час, досягти плавної зупинки, високого гальмівного моменту. Недолік такого способу полягає в тому, що енергія виділяється

в перетворювачі, і в разі швидкої зупинки або великого моменту інерції навантаження для недопущення перегрівання вбудованого резистора контуру постійного струму перетворювача потрібне зовнішній опір. Для того щоб виконати гальмування постійним струмом, або іншими словами, динамічне гальмування, з обмотки статора двигуна знімають змінну напругу і на одну чи дві фази подають постійний струм. При цьому магнітне поле спочатку викликати сповільнення, а потім й утримання ротора в нерухомому стані. Перевагою динамічного гальмування є виділення електричної енергії в роторі двигуна, що робить непотрібним використання гальмівного опору, і плавна зупинка. Брак контролю перетворювачем вихідної частоти призводить до невизначеності часу гальмування. Ефективність гальмування в такому разі порівняно з генераторним методом становить 30–40%.

За комбінованого способу гальмування використовують комбінацію двох описаних способів, тобто на змінну складову вихідної напруги перетворювача накладається постійна складова. Такий спосіб гальмування має переваги обох електричних способів гальмування і дає змогу ефективно гальмувати електродвигун за короткий проміжок часу без виділення теплоти в перетворювачі. Завдяки перетворювачу частоти можна задавати темпи розгону і гальмування залежно від виду застосування і кінематики механізму. Отже, можна задавати лінійну, S-подібну, U-подібну та індивідуальну криві розгону і гальмування (рис. 4.22).

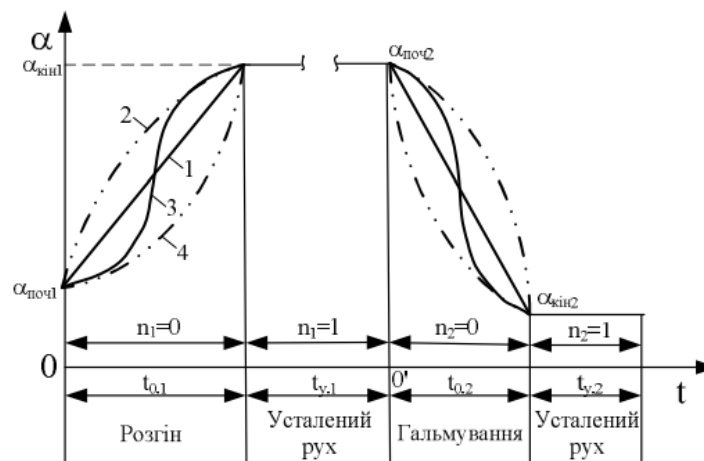


Рис. 4.22. Закони зміни відносної частоти напруги в перехідному процесі АД:
1 – лінійний; 2 – експоненціальний; 3 – S-подібний; 4 – U-подібний (параболічний); $t_{0,1}$ – час розгону; $t_{y,1}$ – час усталеного руху

S-подібний спосіб гальмування/розгону застосовують у конвеєрах, ліфтах. Такий темп розгону дає змогу усунути механічний зазор, унеможливити удари та обмежити розбіжність швидкості із заданою під час швидких перехідних процесів у разі великого моменту інерції. Використання U-подібних кривих розгону/гальмування призначено для установок з відцентровими насосами та зворотними клапанами. Такий темп розгону покращує контроль за роботою клапана.

Часто за частоти живлення в діапазоні 20–40 Гц робота двигунів може бути нестабільною, що призводить до появи коливань струму та напруги. Такі частоти називають забороненими, для того щоби їх компенсувати, потрібно визначити ділянку нестабільності. Індикацією налаштування може слугувати струм двигуна, який не має стрибків, або мінімізація механічних коливань приводу.

Ширину смуги для заборонених частот можна визначити за формулою

$$\Delta f[\text{Гц}] = f_e[\text{Гц}] \frac{Cx[\%]}{100\%}, \quad (4.8)$$

де f_e – заборонена частота; Cx – ширина смуги навколо забороненої частоти.

Пропуск частот, за яких робота електропривода є небажаною, наприклад, через ризик виникнення механічних резонансів, особливо важливий для приводів відцентрових насосів і вентиляторів, в яких явище резонансу виникає особливо часто. Графік зміни частоти промислового перетворювача, представлений на рис. 4.23, ілюструє пропускання «вікон» під час розгону двигуна. Таких «вікон» може бути декілька, їхня ширина зазвичай становить 5 Гц.

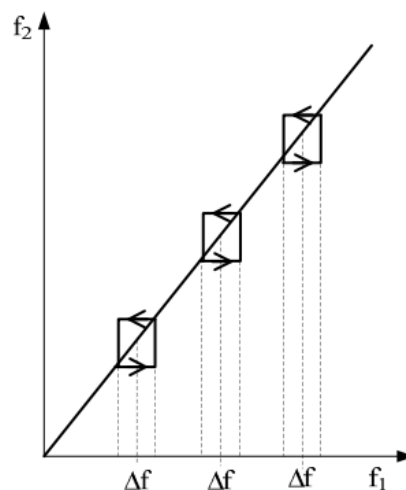


Рис. 4.23. Заборонені частоти та їхні смуги

Для керування кроковим електричним двигуном на його обмотки потрібно подавати імпульсне живлення. На рис. 4.24 наведено осцилограми керування трифазним кроковим двигуном залежно від виду управління.

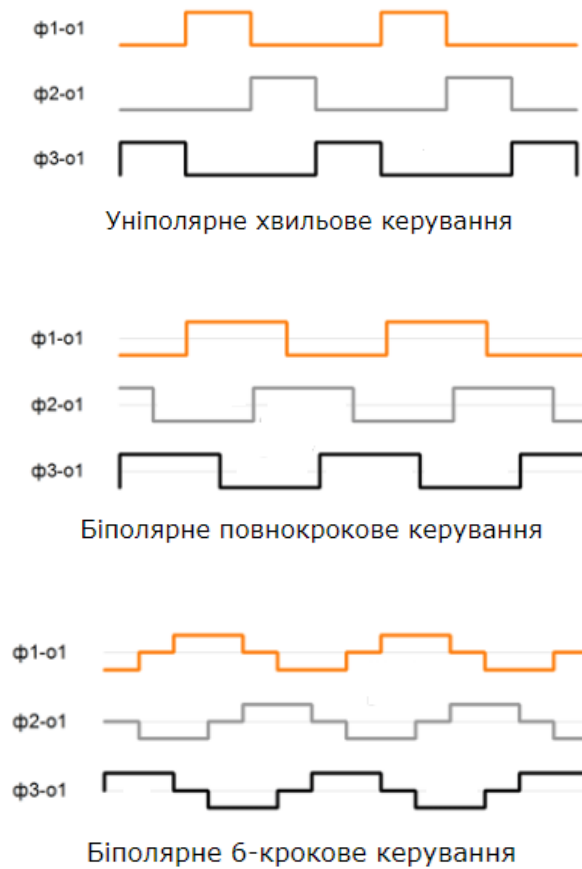


Рис. 4.24. Сигнали управління залежно від способів керування кроковим трифазним електродвигуном

Осцилограму керування для чотирифазного крокового двигуна показано на рис. 4.25.



Рис. 4.25. Сигнали керування кроковим чотирифазним двигуном

Послідовне ввімкнення фаз статора створює магнітне поле, що обертається, за яким слідує ротор. Однак через те що ротор має менше полюсів, ніж статор, ротор повертається за один крок на кут, менший за кут статора. Для реактивного крокового двигуна кут кроку дорівнює:

$$\Theta = \frac{360^\circ}{N_R} - \frac{360^\circ}{N_S}, \quad (4.9)$$

де N_R – кількість полюсів ротора; N_S – кількість полюсів статора.

Щоб змінити напрямок обертання ротора (реверс) реактивного крокового двигуна, треба змінити схему комутації обмоток статора, оскільки зміна полярності імпульсу не змінює напрямки сил, що діють на збуджений ротор.

Реактивні крокові двигуни застосовують тільки тоді, коли потрібен невеликий момент і досить великий крок кута повороту. Такі двигуни тепер рідко використовують.

Для керування кроковим двигуном на постійних магнітах до його обмоток прикладають змінний сфазований струм, який майже завжди має прямокутну форму сигналу, згенерований від джерела постійного струму. Для керування кроковим двигуном з постійними магнітами застосовують декілька способів керування:

- хвильове;
- повнокрокове;
- напівкрокове.

Найпростішим способом керування кроковим двигуном є **хвильове керування**, коли послідовно вмикається лише одна обмотка двигуна. За такого способу керування не досягають максимально можливого моменту.

На рис. 4.26 наведено схеми імпульсів сигналу керування на обмотках управління залежно від конструкції крокових двигунів.

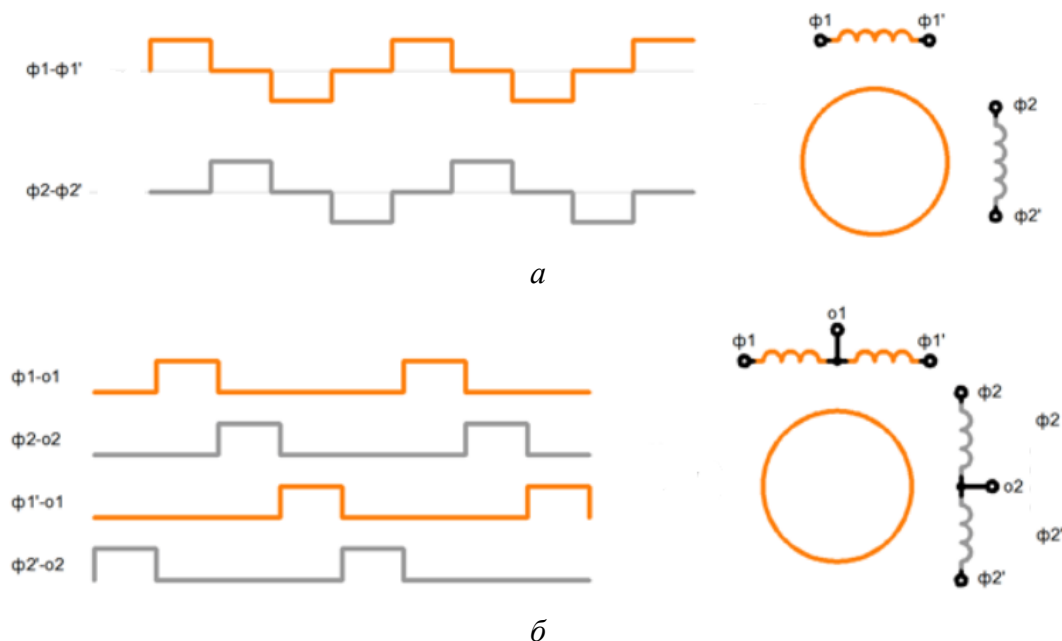


Рис. 4.26. Хвильова схема керування кроковим двигуном:
а – біполярним; б – уніполярним

Повнокрокове керування забезпечує більший момент, ніж хвильове, оскільки обидві обмотки двигуна ввімкнено одночасно. Положення ротора за повнокрокового управління та сигнал керування показано на рис. 4.27.

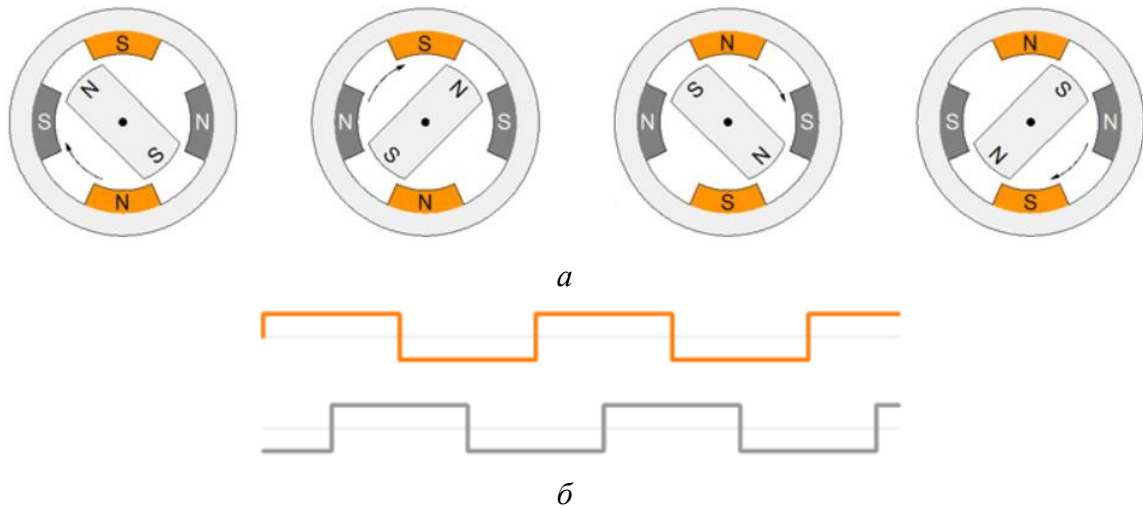


Рис. 4.27. Повнокрокове біполярне керування кроковим двигуном:
a – положення ротора; *б* – сигнал керування

Напівкрокове керування – комбінація хвильового керування та повнокрокового керування з живленням по черзі (рис. 4.28): спочатку однієї обмотки, потім з живленням обох обмоток. За такого управління кількість кроків збільшується вдвічі порівняно з іншими методами керування.

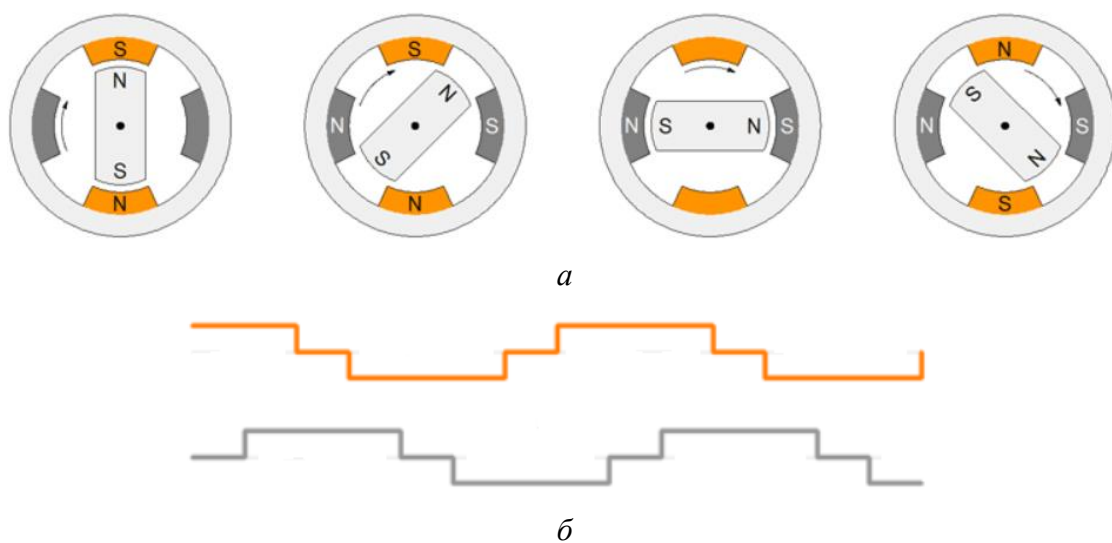


Рис. 4.28. Напівкрокове біполярне керування кроковим двигуном:
a – положення ротора; *б* – сигнал керування

Класична логіка, яка розвивається з давніх часів і використовується в багатьох схемах керування, оснований на булевій алгебрі, де сигнали керування відображаються у вигляді двох станів системи – високий рівень сигналу (ІСТИНА або «1») та низький рівень сигналу (НЕ ІСТИНА або «0») без будь-яких проміжних значень. Все це добре для обчислювальних машин, але управляти реальними системами таким чином досить складно. Розв'язати цю проблему може нечітка логіка.

Нечітка логіка (фазі-логіка) використовує багаторівневе представлення фізичних величин і тверджень, коли окрім крайніх двох рівнів є й проміжні, при цьому перехід від одного рівня до іншого не різкий, а з перекриттям сусідніх рівнів. Для певних фізичних величин і фактів їхнє дворівневе класичне представлення «правда–неправда», цілком справедливе й точне. Наприклад, на запитання, чи рухається робочий орган робота, можна відповісти тільки однозначно «так» або «ні». Але на запитання, як рухається робочий орган (швидко, повільно, дуже повільно), дворівнева логіка не дає відповіді, натомість фазі-логіка дає змогу його одержати. Виділення рівнів у логіці – це асоціативна властивість мислення людини, коли якісна характеристика фізичних величин і процесів, формована словесно (прикметниками, прислівниками), оцінюється кількісно не окремими числами, а деякими числовими величинами з нерізкими межами. Фундаментальним поняттям фазі-логіки є фазі-множина. У математичній логіці є загальне поняття множини як сукупності об'єктів, кожен з яких повинен мати або не мати певну властивість. Ця множина характеризується тільки одним показником, усі об'єкти рівноцінні щодо його визначальної властивості. На відміну від цього поняття множини фазі-множина характеризується двома показниками, по-перше, фактом належності об'єктів до множини й, по-друге, ступенем їхньої належності до цієї множини.

Стосовно технічних систем, об'єктами фазі-множини є значення деякої фізичної змінної, наприклад, значення температури, швидкості переміщення, електричної напруги, струму. Властивістю, що поєднує значення фізичної змінної у фазі-множину, є деяка якісна оцінка в лінгвістичній формі для логічної змінної, наприклад, від'ємна мала (NS), нульова (Z), позитивна середня (PM), додатна велика (PB) тощо. Такі якісні оцінки фазі-множини називаються термами. Ступінь належності значень фізичної змінної до певного терма визначається так званою функцією належності $\mu(x)$. Значення функції належності визначаються в

межах від 0 до 1 на інтервалі від $x = a$, де $\mu(a) = 0$, до $x = b$, де $\mu(b) = 0$. Посередині інтервалу $[a, b]$ є точка $x = c$, де $\mu(c) = 1$ (рис. 4.29).

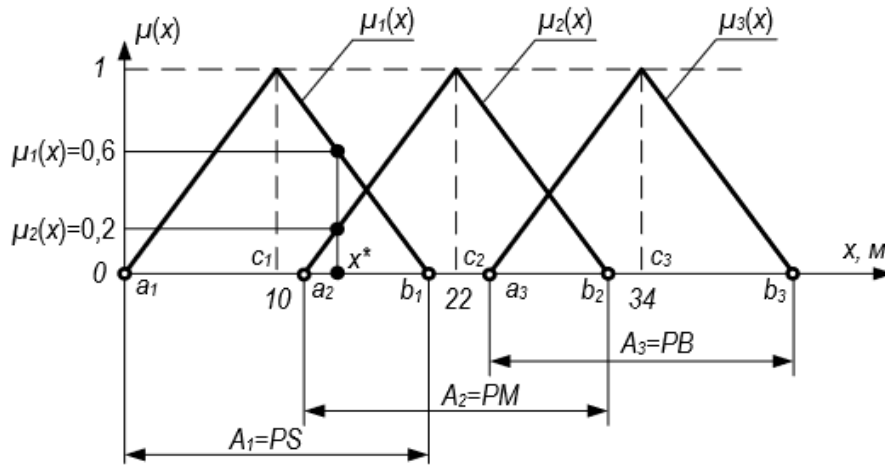


Рис. 4.29. Приклад фазі-множин та функцій належності

На рис. 4.29 представлена термами функція переміщення: додатна мала (PS) для $x = 0 \dots 20$ м; додатна середня (PM) для $x = 12 \dots 32$ м; додатна велика (PB) для $x = 24 \dots 44$ м. Виділеним фазовим множинам A_1, A_2, A_3 відповідними є функції належності $\mu_1(x), \mu_2(x), \mu_3(x)$ із центрами c_1, c_2, c_3 , у яких $\mu(x) = 1$. Сама суть фазі-логіки в її нетвердості й наближеності зумовлює наявність ділянок перекриття термів, де порушується однозначність належності значень змінної x тільки одному терму. Ширина ділянок перекриття може бути різною, навіть нульовою, але максимальна ширина обмежується очевидною умовою: там, де для одного терма $\mu_i(x) = 1$, для іншого може бути тільки $1 > \mu_{i+1}(x) > 0$ та $1 > \mu_{i-1}(x) > 0$. У побудові функцій належності пріоритетного значення набуває розміщення центрів функцій належності, де є $\mu_i(c_i) = 1$, що означає безперечну (відсоткову) належність значення $x = c_i$ до відповідного терму. Якщо $x = x^*$, відстань переміщення належить одночасно до двох термів PS і PM, тому виникає проблема, яке із двох значень функції належності слід прийняти. Ця проблема вирішується компромісно на основі принципів сполучення множин, відомих з математичної логіки, у формі трьох процедур:

– усереднення, або кон'юнкції, $K(\mu_1, \mu_2) = \mu_1 \wedge \mu_2 = \min(\mu_1, \mu_2)$ – принцип мінімуму;

- об'єднання, або диз'юнкції, $D(\mu_1, \mu_2) = \mu_1 \vee \mu_2 = \max(\mu_1, \mu_2)$ – принцип максимуму;
- інверсії або доповнення $\bar{\mu}(x) = 1 - \mu(x)$.

Контрольні запитання

1. У чому сутність управління за критерієм мінімуму середньоквадратичного відхилення?
2. У чому сутність управління за критерієм обмеження порогового значення?
3. Яким чином можна керувати безколекторним електродвигуном?
4. Яким чином відбувається керування частотним перетворювачем?
5. Як відбувається керування кроковим біполярним двигуном з постійними магнітами?
6. Як відбувається управління уніполярним кроковим двигуном з постійними магнітами?
7. В чому відмінність між хвильовим, повнокроковим та напівкроковим керуванням?
8. У чому суть фазі-логіки?

Лекція 5. Пристрої мехатронних систем в логістиці

Для визначення параметрів механічної системи або виявлення змін в навколишньому середовищі та формування інформації про стан вимірних даних і передавання їх в систему керування застосовують **датчики**, або **давачі**. Термін **сенсор** можна вважати синонімом терміна «давач», і особливо великих відмінностей в тлумаченні цього визначення для таких пристроїв немає. Залежно від ступеня обробки сигналу розрізняють такі поняття, як «простий датчик» (його можна вважати сенсором або перетворювачем), «інтегральний датчик» та «інтелектуальний датчик». На рис. 5.1 наведено схему перетворення сигналу із застосуванням найбільш поширених електричних перетворювачів.

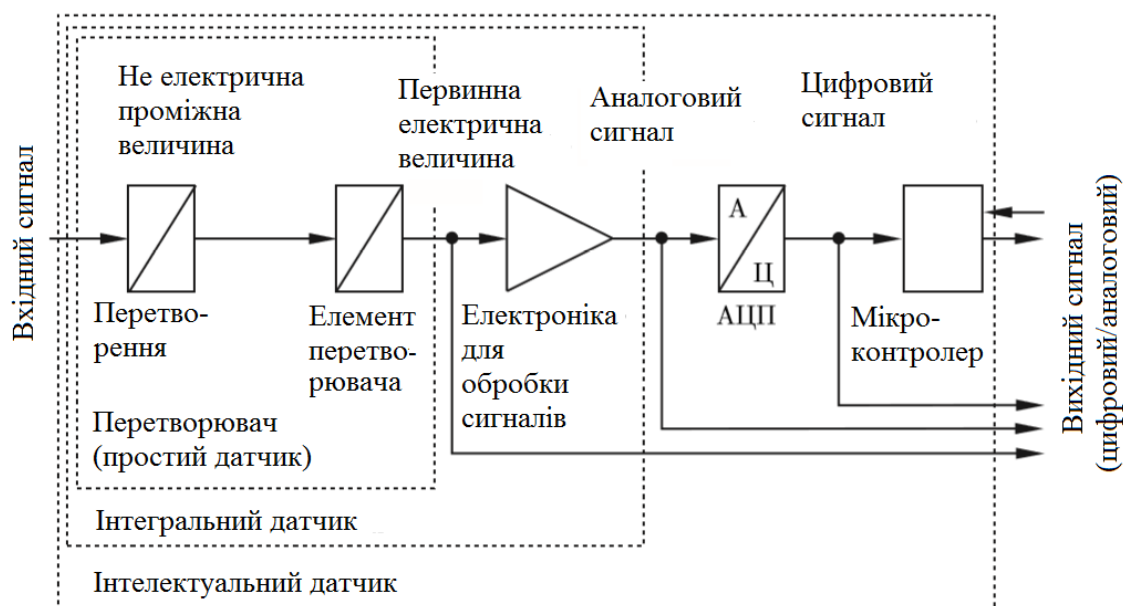


Рис. 5.1. Ступінь інтеграції датчиків залежно від виду обробки сигналу

За характером вихідного сигналу датчики поділяють на дискретні та аналогові. За методом вимірювань (виду вхідних величин) датчики бувають активні (генераторні) та пасивні (параметричні). У параметричних датчиках контрольована величина зумовлена зміною таких параметрів, як активний опір, індуктивність або ємність. Параметричні датчики потребують джерела живлення для виявлення зміни контрольованої величини. У генераторних датчиках зміни контрольованої величини перетворюються в зміну електрорушійної сили на виході датчика, і це не потребує окремого джерела живлення. До основних характеристик датчиків належать: статична характеристика,

чутливість, межа чутливості, похибка, динамічна характеристика, вихідна потужність, вихідний опір, діапазон вимірювань.

Статична характеристика – це залежність вихідної величини від вхідної, може бути лінійною або нелінійною. Для вимірювальних систем з невеликою нелінійною складовою можна виконати приблизну лінеаризацію в діапазоні вимірювання. Для цього нелінійну характеристичну криву наближають до регресійної прямої за методом мінімуму середньоквадратичної похибки.

Чутливість K будь-якого датчика визначається його статичною характеристикою і дорівнює відношенню зміни вихідного сигналу ΔY до зміни вхідної змінної ΔX :

$$K = \Delta Y / \Delta X . \quad (5.1)$$

Діапазон вимірювання – це діапазон вхідних величин, які потрібно відтворити на допустимий діапазон вихідних значень. На практиці треба обирати діапазон вимірювання датчика завжди на кілька відсотків більший, ніж діапазон, що реєструється, щоб перевищення діапазону вимірювання могло бути детектоване і датчик був захищеним від перевантаження.

Статичні похибки вимірювання – це відхилення фактичного вимірюваного значення від очікуваного на основі вхідної величини «ідеального» вимірюваного значення, яке при цьому встановлюють або за допомогою даних виробника, або шляхом власних базових вимірів (калібрування).

Поширеним методом вимірювання переміщення та кута повороту об'єктів, який використовують в різних системах автоматики, є потенціометричний метод, оснований на принципі зміни опору подільника напруги. Таку методику застосовано в механічних потенціометрах шляхом переміщення рухомого контакту (щітки). *Потенціометричні датчики* переміщення бувають лінійними та кутовими (рис. 5.2).

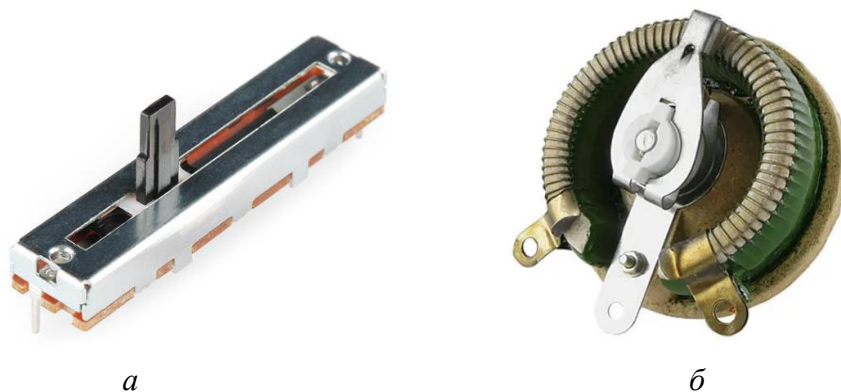


Рис. 5.2. Лінійний (а) та кутовий (б) потенціометри

Регульований резистор R_p потенціометра виконують з тонкого дроту (товщина – до 0,01 мм), шару напівпровідника або металевої плівки. Характеристики датчиків виражаються залежністю $R(x)$ або $U(x)$, де R – вихідний опір, U – вихідна напруга, x – переміщення щітки. На рис. 5.3 наведено варіант розрахункової схеми потенціометричного датчика.

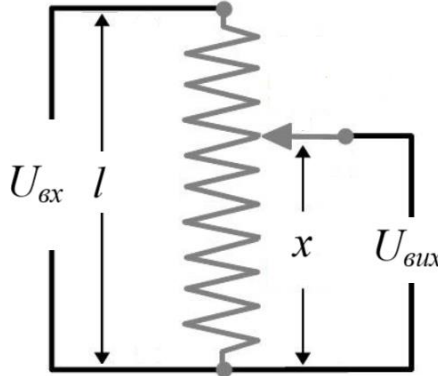


Рис. 5.3. Схема потенціометричного датчика

Якщо опір всієї обмотки датчика позначити як R , а опір її частини, з якої знімається вихідна напруга, як R_x , тоді потенціометрична схема ввімкнення датчика може бути представлена як послідовне з'єднання резисторів з опором R_x і $(R-R_x)$. Струм, що протікає через обмотку датчика, дорівнюватиме $I = U / R$, а прикладена напруга $U_{вх}$ розподілятиметься між послідовно з'єднаними резисторами:

$$U_{вх} = IR_x + I(R - R_x). \quad (5.2)$$

Якщо опір обмотки рівномірно розподілити по всій її довжині l , а переміщення щітки позначити через x , тоді вихідна напруга датчика буде

$$U_{вих} = IR_x = U_{вх} \frac{x}{l}. \quad (5.3)$$

Вихідний сигнал потенціометричного датчика пропорційний переміщенню його щітки. Для надійного контакту між рухомим контактом й обмоткою потрібно прикладати велику силу притиснення движка.

Важливою характеристикою потенціометра є плавність зміни його вихідної напруги. Під час переміщення щітки по обмотці потенціометра відбувається стрибкоподібна зміна опору, що приводить до стрибкоподібної зміни вихідної напруги. Якщо для лінійного потенціометра взяти за ідеальну характеристику пряму, що проходить

через середини стрибків, тоді зона нечутливості або абсолютна похибка в одиницях опору становитиме:

$$\Delta R_c = \pm R / 2\omega, \quad (5.4)$$

де R – максимальний опір потенціометра; ω – повна кількість витків.

Тахогенератор – це електричний вимірювальний генератор постійного або змінного струму, призначений для перетворення миттєвого значення частоти (кутової швидкості) обертання вала в однозначно пов'язаний зі швидкістю електричний сигнал (за напругою або частотою). Електричний сигнал тахогенератора подається або для безпосереднього відображення та зчитування показів на проградуирований в одиницях швидкості обертання вторинний прилад – індикатор тахометра, або на вхід автоматичних пристроїв керування, що регулюють частоту обертання.

За принципом дії тахогенератори поділяють на пристрої з вихідним сигналом змінного струму або напруги (синхронні та асинхронні) і вихідним сигналом постійного струму.

Тахогенератори постійного струму – це невеликі колекторні машини, потік збудження яких створюється постійним магнітом або незалежною обмоткою. Ці тахогенератори є звичайними колекторними генераторами постійного струму, але з постійним збудженням, зазвичай створюваним постійними магнітами статора (рис. 5.4).

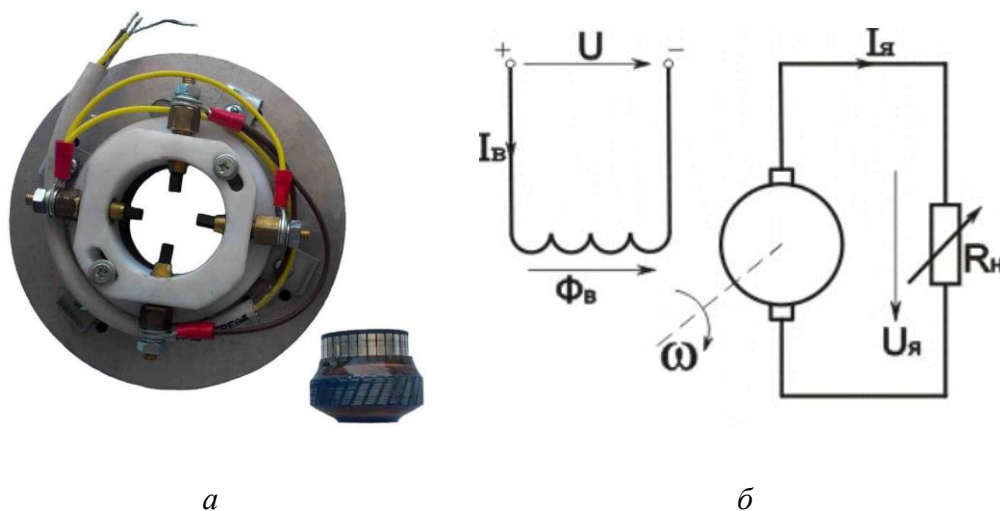


Рис. 5.4. Зовнішній вигляд тахогенератор МТ-6 (а) та схема його конструкції (б)

Характеристику тахогенератора постійного струму за холостого ходу, тобто за розімкнутого ланцюга якоря, наведено на рис. 5.5.

Асинхронні тахогенератори змінного струму конструктивно подібні до асинхронних електродвигунів з короткозамкненим ротором, зазвичай у вигляді порожнистого алюмінієвого або мідного циліндра. На статорі такого тахогенератора з магнітними потоками, орієнтованими один на одного під кутом 90° , розміщено дві обмотки, одна з яких (обмотка збудження) живиться змінним синусоїдальним струмом постійної частоти і постійної амплітуди, а друга є вихідною обмоткою, і до неї може бути приєднаний вимірювальний прилад (відградуйований вольтметр змінної напруги, наприклад, в об/хв) або вхід автоматичної системи управління (рис. 5.6). Асинхронний тахогенератор дає змогу визначати напрямок обертання ротора, оскільки після зміни напрямку обертання фаза вихідного сигналу змінюється на 180° .

Синхронні тахогенератори змінного струму – це безколекторна синхронна машина з ротором, який підмагнічений постійним магнітом. На статорі розміщено одну або кілька обмоток. Такий тахогенератор перетворює швидкість обертання ротора на змінну напругу, амплітуда і частота якого прямо пропорційні швидкості обертання ротора. Вимірювання швидкості обертання можливе двома способами – частотним й амплітудним. Оскільки частота вихідного сигналу не залежить від температури, зменшення магнітного потоку, спричиненого старінням, і величини зазору між ротором і статором тахогенератора, частотний спосіб є одним з найточніших. Швидкість обертання ротора обчислюється шляхом визначення частоти вихідного сигналу та подальшим обчисленням частоти обертання за формулою

$$n_f = \frac{f_s}{p}, \quad (5.5)$$

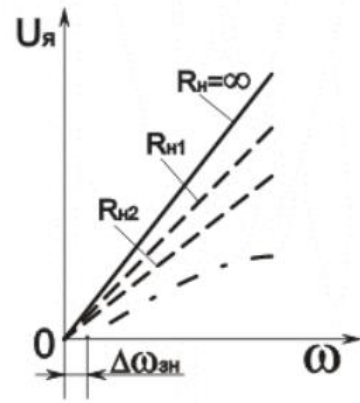


Рис. 5.5. Характеристика тахогенератора постійного струму за холостого ходу

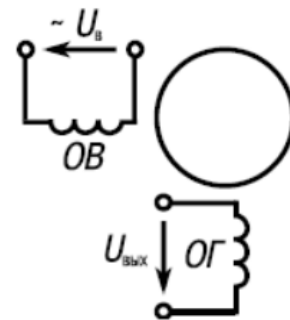


Рис. 5.6. Схема конструкції асинхронного тахогенератора

де n_f – частота обертання ротора, Гц; f_s – частота сигналу на виході тахогенератора, Гц; p – кількість пар полюсів ротора тахогенератора.

Недоліком частотного методу (рис. 5.7) є те, що для більш точного визначення швидкості потрібно більше часу, і за цей час швидкість може значно змінитися. З цього випливає, що чим більше часу витрачається на накопичення імпульсів для визначення частоти, тим більшою є динамічна похибка у вимірюваннях, тому в системах автоматичного регулювання швидкості обертання відбувається із запізненням реакції на обурення.

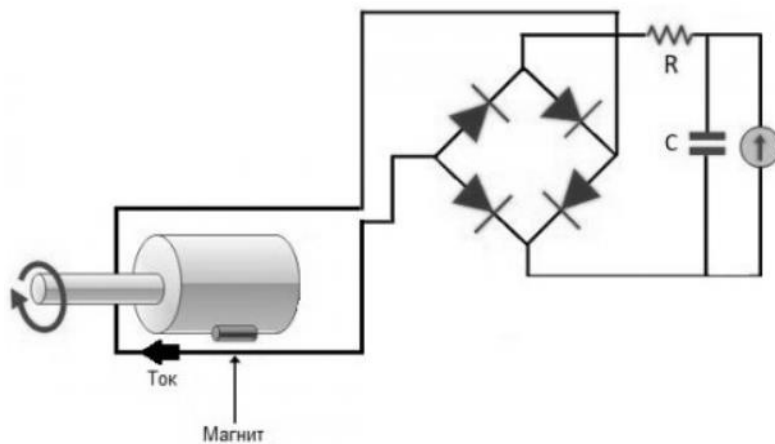


Рис. 5.7. Частотний спосіб визначення швидкості обертання

Для зниження динамічної похибки використовують тахогенератори з більшою кількістю полюсів, це дає змогу зменшити час визначення вихідної частоти, отже, і час реакції системи авторегулювання. Визначити частоту сигналу можна з накопичених та усереднених періодів кількох імпульсів. Розрахунок виконують за формулою

$$f_s = \frac{N_s}{T_1 + \dots + T_N} = \frac{N_s}{\sum_{i=1}^N T_i}, \quad (5.6)$$

де f_s – частота сигналу на виході тахогенератора; N_s – кількість накопичених імпульсів; T – тривалість періоду.

Відношення вихідної напруги до частоти обертання ротора називають «чутливістю тахогенератора», або «коефіцієнтом перетворення», або «крутизною тахогенератора» і зазвичай вказують у технічній специфікації тахогенератора в мілівольтах на оборот за хвилину.

Амплітудний спосіб визначення частоти обертання ротора за цим параметром та вихідною напругою полягає в такому:

$$n = \frac{U}{S}, \quad (5.7)$$

де n – частота обертання ротора в оборотах за хвилину; U – вихідна напруга тахогенератора; S – коефіцієнт перетворення.

Датчик кута повороту, або *енкодер*, – кодовий вимірювальний перетворювач кута повороту в цифрові або аналогові сигнали.

Датчики кута повороту поділяються так:

- за способом видачі інформації – на накопичувальні (інкрементні) та абсолютні (позиційні);
- за принципом дії – на оптичні, резистивні, магнітні, індуктивні, механічні;
- за допустимим кутом повороту вала – на пристрої з обмеженим діапазоном роботи та пристрої з необмеженим діапазоном роботи.

Накопичувальні енкодери (рис. 5.8) на виході формують імпульси, за якими приймальний пристрій визначає поточне положення вала шляхом підрахунку кількості імпульсів.

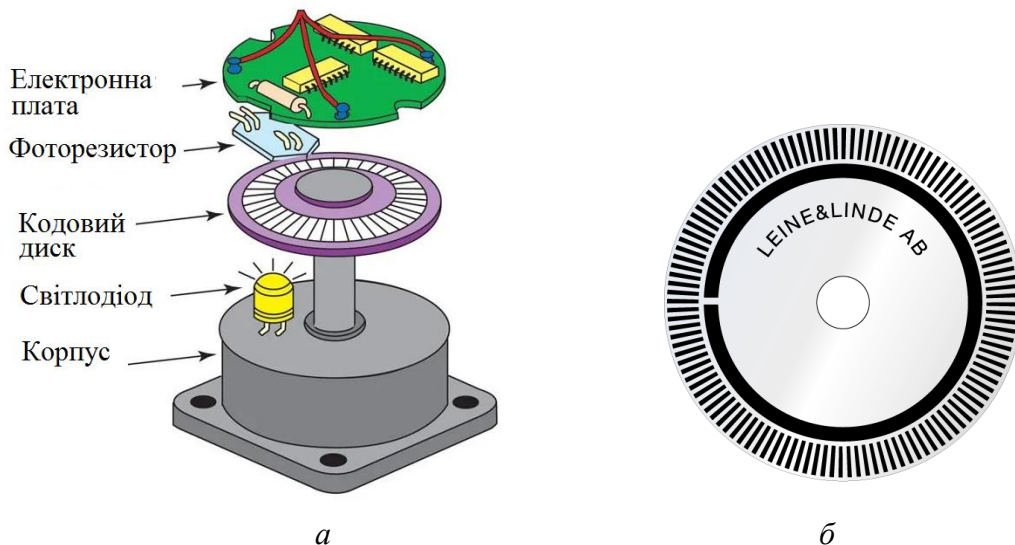


Рис. 5.8. Конструкція оптичного інкрементного енкодера (а) та кодового диска (б)

Після ввімкнення накопичувального енкодера положення вала буде невідоме. Для прив'язки системи відліку до початку відліку накопичувальні датчики мають нульові (референтні) мітки, які диску потрібно пройти після ввімкнення обладнання. Основним недоліком такого типу датчиків кута положення є брак механізмів визначення пропусків імпульсів від енкодера, що призводить до накопичення помилки у визначенні кута повороту вала, доки не буде пройдена нуль-

мітка. Для визначення напрямку обертання застосовують два вимірювальні канали, в яких ідентичні послідовності імпульсів (меандр) зміщені на 90° одна відносно одної (рис. 5.9).

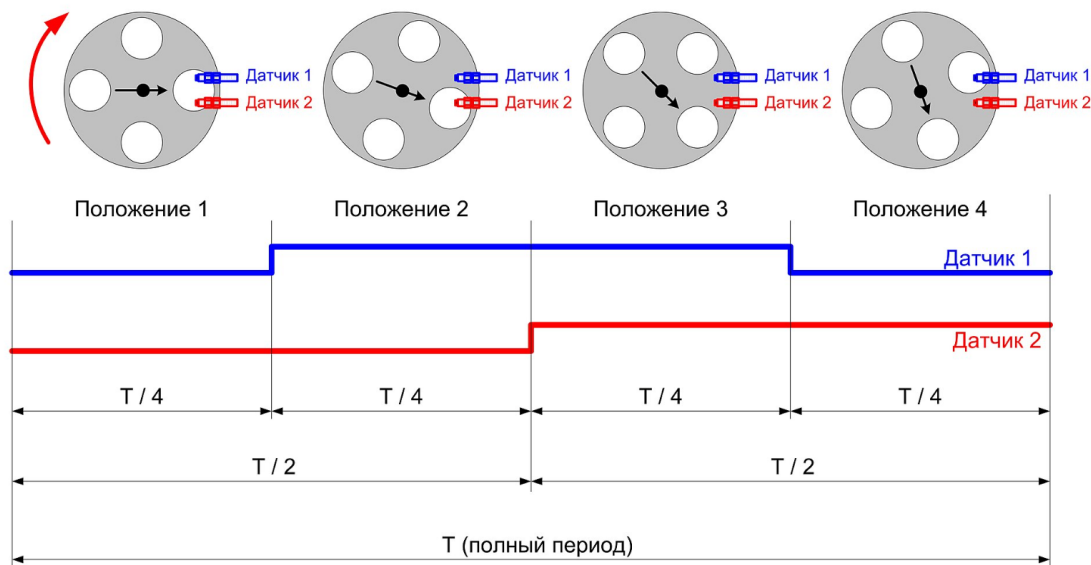


Рис. 5.9. Принцип роботи накопичувального датчика кута повороту

Абсолютні енкодери формують на виході сигнали, які однозначно можна інтерпретувати, як кут повороту вала (рис. 5.10) датчика. Енкодери цього типу не потребують прив'язки системи відліку до будь-якого нульового положення. Перевагою абсолютного енкодера є те, що він стійкий проти впливу. У випадках, коли сигнал з диска не може бути повноцінно прочитаний енкодером, наприклад, якщо вал обертається занадто швидко, точний кут повороту буде зареєстрований, коли швидкість обертання зменшиться. Проте навіть за швидкого обертання вала можлива реєстрація його положення, хоча й з меншою точністю.

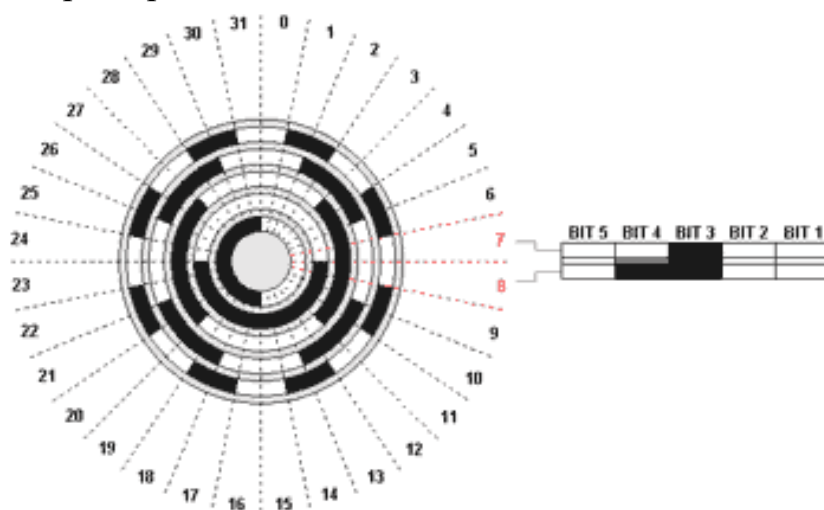


Рис. 5.10. Конструкція диска абсолютного енкодера з растром у коді Грея

Бінарні коди – це коди, які складаються тільки з двох символічних станів, наприклад, чорний або білий, світлий або темний, провідник або ізолятор. Бінарний код у цифровій техніці – це спосіб представлення даних (чисел, слів тощо) у вигляді комбінації двох знаків, які можна позначити як «0» або «1». Знаки, чи одиниці, таких кодів називають бітами. Однією з переваг застосування бінарних кодів є простота та надійність накопичення інформації на носії у вигляді комбінації всього двох його фізичних станів, наприклад у вигляді зміни або сталості світлового потоку під час зчитування з оптичного кодового диска. У табл. 5.1 представлено дві основні можливості кодування – двійкове кодування та кодування за методом Грея, а також способи подання інформації в різних системах числення.

Таблиця 5.1

Таблиця з бінарними кодами

Звичайне двійкове кодування			Кодування у коді Грея		
DEC	BCD (Binary)	HEX	DEC	BCD (Gray)	HEX
0	0000	0h	0	0000	0h
1	0001	1h	1	0001	1h
2	0010	2h	2	0011	2h
3	0011	3h	3	0010	3h
4	0100	4h	4	0110	4h
5	0101	5h	5	0111	5h
6	0110	6h	6	0101	6h
7	0111	7h	7	0100	7h
8	1000	8h	8	1100	8h
9	1001	9h	9	1101	9h
10	1010	Ah	10	1111	Ah
11	1011	Bh	11	1110	Bh
12	1100	Ch	12	1010	Ch
13	1101	Dh	13	1011	Dh
14	1110	Eh	14	1001	Eh
15	1111	Fh	15	1000	Fh

Для різних систем числення в таблиці використано такі позначення:

- десяткове число → decimal → **DEC**;
- двійково-десяткове число → binary code decimal → **BCD**;
- шістнадцякове число = hexadecimal = **HEX**.

Код Грея – одна з систем кодування інформації, у якій два послідовні коди відрізняються значенням лише одного біта. Код Грея є

кращим за звичайний двійковий тим, що має властивість сталості бінарної комбінації: зміна кодованого числа на одиницю відповідає зміні кодової комбінації тільки в одному розряді. Грей-код є так званим однокроковим кодом. Він будується на базі двійкового за таким правилом: старший розряд залишається без зміни, а кожен наступний розряд інвертується, якщо попередній розряд вихідного двійкового коду дорівнює одиниці.

Тензометричні датчики використовують для вимірювання переміщень, деформацій або зусиль. Принцип роботи тензометричних датчиків ґрунтується на явищі тензоефекту – коли зміна величини активного опору провідникового або напівпровідникового матеріалу змінюється під дією прикладених механічних напружень.

На рис. 5.11 наведено конструкцію тензорезисторів – основного перетворювача тензометричного датчика.

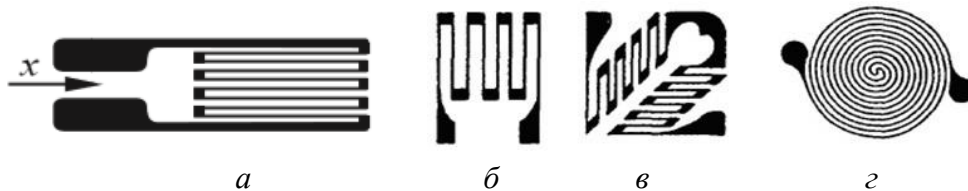


Рис. 5.11. Будова тензорезисторів: *а* – дротовий тензодатчик; *б* – фольговий тензорезистор для виміру лінійних переміщень; *в* – фольговий тензорезистор для вимірювання деформацій у двох взаємно перпендикулярних напрямках; *г* – фольговий тензорезистор для вимірювання тиску

Найпростішим виконанням тензорезистора є провідник з матеріалу високого питомого опору і малим температурним коефіцієнтом, наліплений у вигляді розмітки на основу плівки. В подальшому такий тензорезистор приклеюють до випробовуваної деталі, завдяки чому деформацію деталі сприймають дротові ґрати. Довжина деталі, займана дротом, називається вимірювальною базою датчика l .

Чутливість тензорезистора визначається коефіцієнтом відносної тензочутливості:

$$S = \frac{\Delta R/R}{\Delta l/l} = E \frac{\Delta R}{R\sigma}, \quad (5.8)$$

де $\Delta R, \Delta l$ – приріст опору та довжини; R, l – опір та довжина провідника; σ – механічне напруження в матеріалі; E – модуль пружності.

Через незначну зміну опору одного тензорезистора внаслідок його деформації, застосовують набір тензорезисторів, які з'єднують за принципом мостової схеми Вітстона (рис. 5.12).

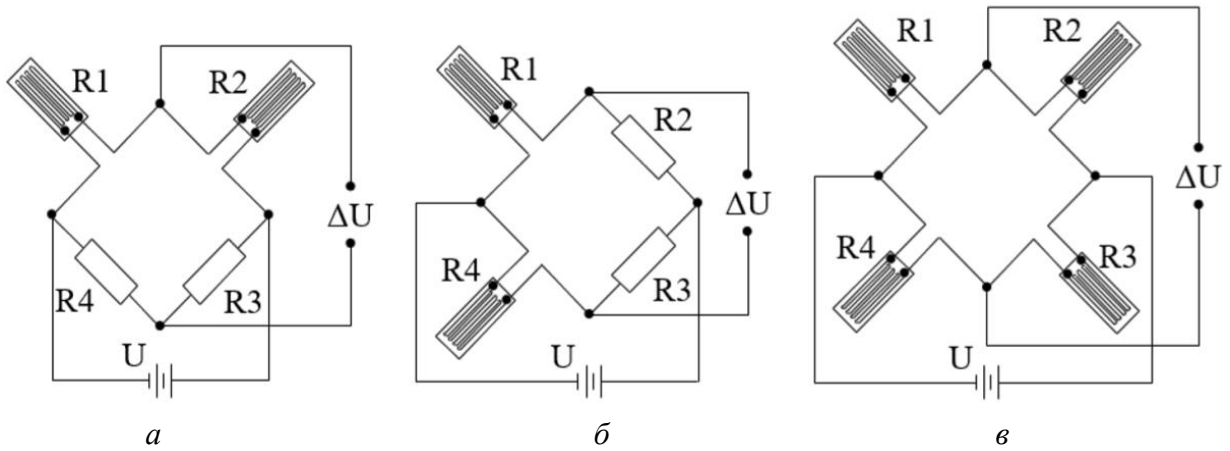


Рис. 5.12. Мостові схеми під'єднання тензорезисторів: *a, б* – напівмостова; *в* – повномостова (схема Вітсона)

Мостова схема під'єднання тензорезисторів обов'язково повинна бути збалансованою, тобто

$$\frac{R_1}{R_2} = \frac{R_3}{R_4}, \quad (5.9)$$

де R_1, R_2, R_3, R_4 – опори резисторів мостової схеми.

Вихідна напруга вимірювального моста (рис. 5.12) дорівнює:

$$U_0 = \left[\frac{R_3}{R_3 + R_4} - \frac{R_2}{R_1 + R_2} \right] U, \quad (5.10)$$

де U – напруга живлення схеми, В.

Для того щоб спотворення вихідної напруги U_0 від під'єднаної електроніки були незначними, її вхідний опір має бути дуже великим порівняно з номінальним опором тензодатчиків, а для цього часто використовують тензопідсилювач (5.13). Якщо обидва плеча моста під'єднати до входу операційного підсилювача з вхідними каскадами, тоді можуть бути реалізовані вхідні опори порядку 10^{12} Ом. Перевагою операційного підсилювача є можливість налаштування через єдиний контрольований опір R_1 .

Робота *ємнісних датчиків* полягає в перетворенні вимірюваної величини в ємнісний опір (рис. 5.14). Ємність плоского конденсатора:

$$C = \varepsilon \varepsilon_0 S / d, \quad (5.11)$$

де ε – відносна діелектрична проникність середовища між обкладками;
 ε_0 – діелектрична постійна ($\varepsilon_0 = 8,85 \cdot 10^{-12}$ Ф/м); s – площа обкладок;
 d – відстань між обкладками.

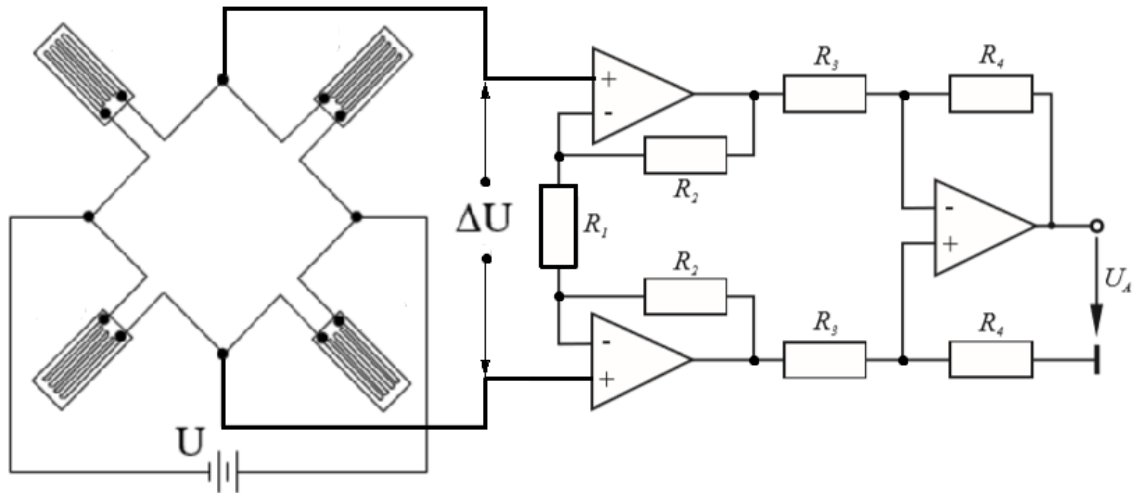


Рис. 5.13. Схема підключення тензопідсилювача

Зміна ємності конденсатора може відбуватися через зміну кожної з трьох величин: d , s , ε . Часто такі типи датчиків використовують для вимірювання прискорення (акселерометри), вібрації, нахилу основи, на якій їх розміщено, вологості.

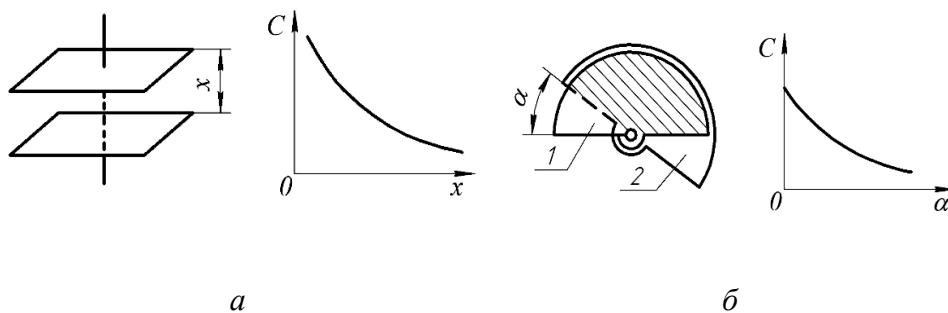


Рис. 5.14. Схеми та характеристики лінійного (а) та кутового (б) датчиків

Датчик Холла застосовують для вимірювання напруженості магнітного поля. Більшість промислових датчиків Холла виготовляють на базі напівпровідників, принцип дії яких такий: якщо пластину напівпровідника помістити у магнітне поле з напруженістю H , а по пластині тече струм величиною I і при цьому лінії напруженості електричного поля спрямовані під прямим кутом з напрямком протікання струму, тоді на бокових гранях цієї пластини виникає напруга (рис. 5.15).

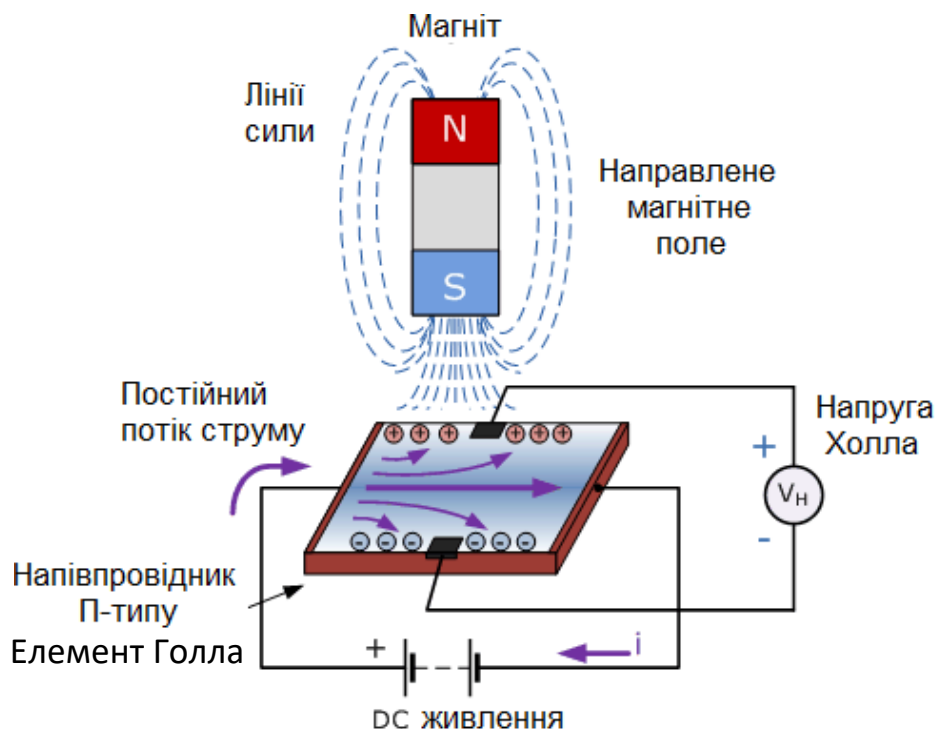


Рис. 5.15. Принцип роботи датчика Голла

Вихідний сигнал датчика:

$$U_{вих} = E = K \cdot I \cdot H, \quad (5.12)$$

де $K = k_x / d$ – коефіцієнт, що залежить від матеріалу і товщини пластини d ; k_x – постійна Голла; I – сила струму, який протікає по пластині; H – напруженість магнітного поля.

П'єзоелектричні датчики працюють на основі явища п'єзоефекту, суть якого полягає в тому, що під дією прикладених зусиль на гранях деяких кристалів (кварц, титан, турмалін) з'являється електричний заряд (прямий п'єзоефект) (рис. 5.16, а). У разі внесення п'єзоелемента в електричне поле він деформується (обернений п'єзоефект).

Одним з недоліків п'єзоелектричних датчиків є те, що їх не можна використовувати для статичних вимірювань. Статична сила призводить до утворення фіксованої кількості заряду на п'єзоелектричному матеріалі.

Вихідний сигнал датчика:

$$U_{вих} = \frac{Qd}{\epsilon f}, \quad (5.13)$$

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

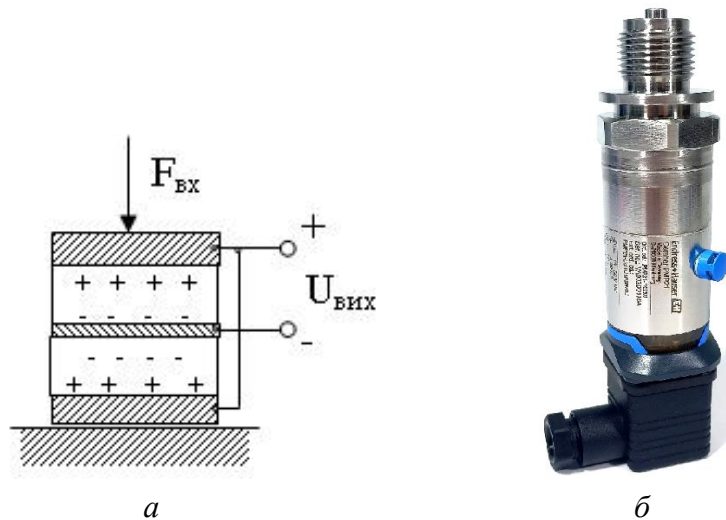


Рис. 5.16. Принцип дії п'єзоелектричного датчика (а) та вигляд датчика тиску на основі п'єзокристала (б)

П'єзоелектрична технологія може вимірювати різні фізичні величини, найчастіше – тиск і прискорення. Для датчиків тиску (рис. 5.16, б) використовують тонку мембрану та масивну основу, яка гарантує, що прикладений тиск навантажує елементи в одному напрямку. У акселерометрах рухоми масу прикріплено до кристалічних елементів, і коли акселерометр реєструє рух, рухома маса навантажує ці елементи відповідно до другого закону руху Ньютона. Основна відмінність у принципі роботи між цими двома випадками полягає в тому, як вони прикладають зусилля до чутливих елементів. У датчику тиску тонка мембрана передає зусилля на чутливий елемент, а в акселерометрі прикріплена маса деформує чутливий елемент.

Вимірювання переміщення за допомогою ультразвуку ґрунтується на вимірюванні часу проходження ультразвукового імпульсу. В ролі ультразвукового перетворювача використовують п'єзоелектричні елементи, які в процесі подачі напруги деформуються, внаслідок чого виникає високочастотне коливання, що складається з небагатьох серій коливань. Зазвичай використовують частоти 20...200 кГц. Під час генерування ультразвукових коливань безпосередньо перед датчиком утворюється простір певної інтенсивності (рис. 5.17), довжина якого приблизно становить:

$$r = \frac{D^2}{4\lambda}, \quad (5.14)$$

де D – діаметр ультразвукового вібратора (генератора); λ – довжина хвилі.

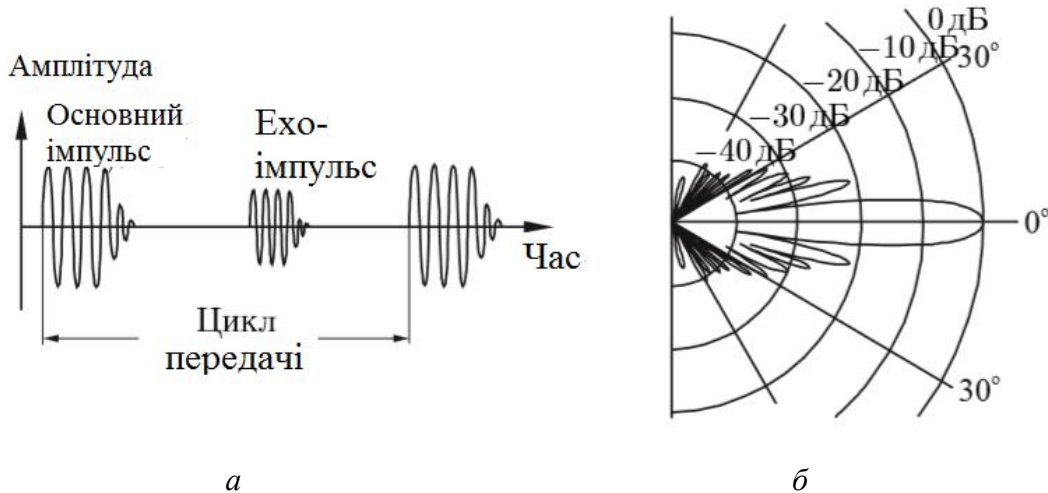


Рис. 5.17. Вид сигналу ультразвукового датчика: a – схема генерування хвиль; b – діаграма поширення

Кут розсіювання ультразвукової хвилі для кварцового генератора:

$$\alpha \approx 2,44 \frac{\lambda}{D}. \quad (5.15)$$

Поширення ультразвукової хвилі в робочому середовищі відбувається за відомим законом хвиль:

$$v = \lambda f. \quad (5.16)$$

де v – швидкість хвилі; λ – довжина хвилі; f – частота хвилі.

Якщо фіксувати інтервал часу між фронтами пускового імпульсу та відлуння (ехо), тоді відстань до перешкоди можна визначити за формулою:

$$H = \frac{\Delta T v}{2} \cos \alpha, \quad (5.17)$$

де v – швидкість хвилі (швидкість звуку в заданому середовищі), ΔT – різниця часу між фронтами пуску та відлуння; α – кут відбиття хвилі (зазвичай від 0 до 30°).

Під час руху об'єкта частота відбитих хвиль не збігається з частотою випромінюваних хвиль (ефект Доплера). У класичній фізиці, де швидкості джерела та приймача відносно середовища є нижчими за швидкість хвиль у середовищі, співвідношення між спостережуваною частотою f і частотою випромінювання f_0 задається формулою

$$f = \left(\frac{c \pm v_r}{c \pm v_s} \right) f_0, \quad (5.18)$$

де c – швидкість поширення хвилі в середовищі; v_r – швидкість приймача відносно середовища (додається, якщо приймач рухається до джерела, віднімається, якщо приймач віддаляється від джерела); v_s – швидкість джерела відносно середовища (додається, якщо джерело віддаляється від приймача, віднімається, якщо джерело рухається до приймача).

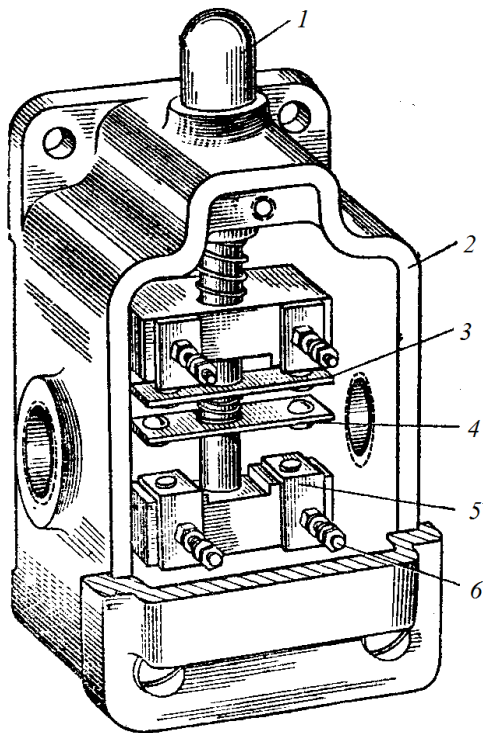


Рис. 5.18. Кінцевий вимикач ВК-411: 1 – стрижень з контактними пластинками; 2 – корпус; 3, 4 – рухомі контакти; 5 – нерухомі контакти; 6 – затискачі

Найпростішим дискретним датчиком контактного типу є кінцевий вимикач, який спрацьовує в разі досягнення механічною частиною пристрою певного положення. Стандартні кінцеві вимикачі в промислових системах керування виготовляють з різними типами робочих механізмів. Це може бути важіль, плунжер або пружина. Кінцеві вимикачі можуть прямо перемикатися робочим важелем з контактами (рис. 5.18).

Геркони спрацьовують, коли повз них проходить магнітне поле магніту, який вбудовано в рухому деталь. Геркон являє собою герметичну скляну колбу, в якій є дві (або

більше) металеві контактні пластини. Всередині колби контакти гнучкі. Пластини геркона розміщено в паралельних площинах на малій відстані одна від одної (рис. 5.19). Виготовляють геркони також з перекидним контактом.



Рис. 5.19. Геркон: а – вигляд; б – схематичне позначення

Безконтактні перемикачі (сенсори) можуть спрацювати внаслідок збурення електромагнітного поля, зміни електричної ємності, появи магнітного поля (наприклад, давачі Голла).

Однією з вимог до інтелектуальної техніки є здатність сприймати навколишнє середовище. *Машинний зір* – це комп'ютерна версія людського зору, коли камери фіксують інформацію, а комп'ютери її оброблюють. Вихідними даними систем зору є цифрові зображення. Цифрове зображення складається з маленьких квадратів, які називаються *пікселями* (елементами зображення) і які містять інформацію про рівень інтенсивності світла. Цифрова або аналогова камери в такій інформаційній системі є оптичним датчиком, який формує піксельне зображення.

Більшість цифрових камер (рис. 5.20), використовуваних на логістичних системах, є CCD-пристроями (із зарядовим зв'язком), які складаються з невеликих прямокутних датчиків, що містять крихітні світлочутливі комірки, кожна з котрих видає інформацію про відповідний піксель зображення.



Рис. 5.20. Цифрові CCD-фотокамери

Якщо зображення чорно-біле (технічно це називається монохромним), то рівень інтенсивності представлений у шкалі сірого між мінімальним значенням (0) і максимальним значенням ($i_{\max}$). Кількість рівнів у шкалі сірого залежить від кількості бітів, у яких закодовано зображення. Більшість зображень, які використовуються, є 8-бітними, це означає, що зображення може розрізняти 256 рівнів сірого, де мінімальне значення 0 – це глибокий чорний колір, а максимальне значення 255 представляє чистий білий. На практиці людське око не може розрізнити стільки рівнів, а тому восьми біт більш, ніж достатньо. Коли цифрові зображення відтворюють кольорову сцену, пікселі передають інформацію про рівні інтенсивності для трьох каналів –

червоного (R), зеленого (G) і синього (B), що дає зображення RGB. Обробка RGB-зображень є складнішою, ніж монохромних.

Для безконтактного вимірювання швидкості можна використовувати оптичні кореляційні системи (рис. 5.21) за допомогою CCD-камер, встановлених на певній відстані, які фіксують два знімки об'єкта, що рухається. Оскільки об'єкт має певну структуру поверхні, він виявляється на обох фото просторово зміщеним. Шляхом кореляційного аналізу можна визначити величину зміщення та з відомої відстані зйомки обчислити швидкість.

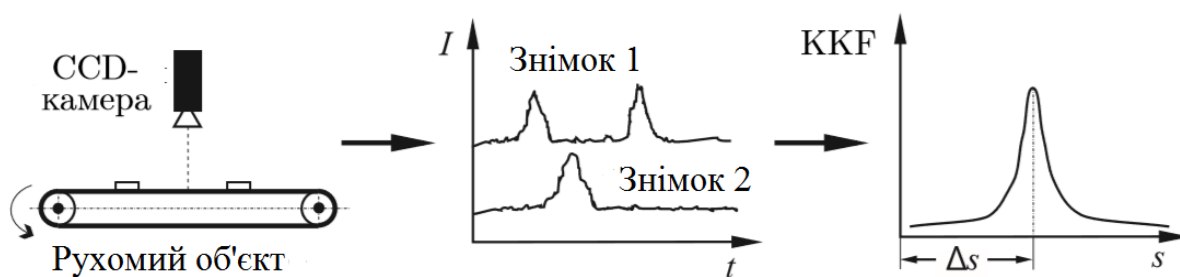


Рис. 5.21. Принцип оптичного кореляційного вимірювання: I – інтенсивність світла; ККФ – функція взаємної кореляції

Монокулярні камери (з одним об'єктивом) є простими системами зору, які дають чітку інформацію про зображення. Важливими технічними параметрами камер є фокусна відстань об'єктива, розмір датчика та оптичні фільтри, коли є спектральні діапазони (кольори), які потрібно заблокувати від зображення. Фокусна відстань пов'язана з масштабом сцени, яка вписується в зображення. Таким чином, конструкція системи машинного зору повинна мати правильні параметри камери та об'єктива, щоб забезпечити потрібний кут зору та попадання цільових об'єктів у фокус об'єктива.

Контрольні запитання

1. Які основні принципи покладено в основу побудови мехатронних систем?
2. Які пристрої можуть бути складовою частиною машин з комп'ютерним керуванням рухом?

Змістовний модуль 2. ЛОГІСТИКА МЕХАТРОННИХ СИСТЕМ

Лекція 6. Основні принципи SCM-логістики

Використання Інтернет-технологій в мехатроніці відкриває нові перспективи розвитку розподілених систем управління, збору та аналізу даних. Завдання дистанційного моніторингу експериментів і контролю віддалених технічних систем за допомогою Інтернету можуть бути виконані з мінімальними витратами практично в будь-якій точці світу завдяки значному поширенню і доступності глобальної мережі. Дистанційне керування мехатронними об'єктами з використанням Інтернету означає не тільки збір даних за допомогою інформаційно-виміральної апаратури, а й керівні впливи на виконавчі елементи різних типів.

Аналіз сучасних тенденцій, а також ряду реально здійснених проєктів свідчить, що Інтернет є досить ефективним і зручним засобом організації дистанційного керування технічними об'єктами, перевагами якого є таке:

- можливість організувати дистанційне керування реальними об'єктами і експериментами практично з будь-якої точки світу;
- зниження витрат на створення спеціалізованих каналів зв'язку;
- можливість організації доступу широкого кола фахівців-експертів, територіально віддалених один від одного, до унікального обладнання в режимі реального часу.

Перспективні сфери застосування Інтернет-мехатронії:

- дистанційне керування в умовах агресивного навколишнього середовища;
- дистанційна освіта (створення віртуальних лабораторій віддаленого доступу);
- віддалене керування.

Для ефективного використання перспектив, які відкриває застосування глобальної мережі у сфері керування технікою, слід розробити такий підхід до її використання, який враховував би проблеми, зумовлені специфікою Інтернету як каналу зв'язку в системі дистанційного керування технічним об'єктом, а також специфіку керованого об'єкта.

Залежно від способів організації обчислювального й інтерфейсного пристрою відповідно змінюється й загальна організація системи комп'ютерного керування.

До системи комп'ютерного керування мехатронним об'єктом можна поставити ряд різноманітних вимог, наприклад:

- вимоги до швидкодії;
- вимоги до надійності;
- вимоги (обмеження) щодо вартості;
- вимоги (обмеження) щодо споживаної потужності;
- вимоги (обмеження) щодо масогабаритних характеристик;
- вимоги щодо стійкості впливу зовнішніх факторів;
- вимоги, що обмежують негативний вплив на зовнішнє середовище.

Під швидкістю системи комп'ютерного керування будемо розуміти величину, обернено пропорційну часу розв'язування задач з керування мехатронним об'єктом. Що менший цей час, то більшою буде швидкість системи комп'ютерного керування. Очевидно, що час на розв'язування завдання комп'ютерного керування складається із часу приймання вихідних даних від інформаційної системи, часу реалізації алгоритму керування й часу передавання команд керування на виконавчу систему.

Надійність – це властивість системи не витрачати з часом здатності виконувати потрібні функції. Очевидно, що ця здатність і залежатиме від здатності інтерфейсної частини передавати дані між інформаційною й виконавчою системами мехатронного об'єкта й обчислювальною частиною системи комп'ютерного керування, а також від здатності обчислювальної частини пристрою системи комп'ютерного керування реалізувати алгоритм керування за певний час. Для кількісної оцінки надійності використовують різні критерії, такі як імовірність безвідмовної роботи, середній наробіток на відмову та інші.

Виділяють три способи організації обчислювальних механізмів системи комп'ютерного керування (рис. 6.1).



Рис. 6.1. Способи організації обчислювального пристрою системи комп'ютерного керування

Однопроцесорна централізована організація обчислювального пристрою системи комп'ютерного керування означає наявність одного процесорного вузла, на який покладається функція реалізації алгоритму керування мехатронного об'єкта (рис. 6.2).

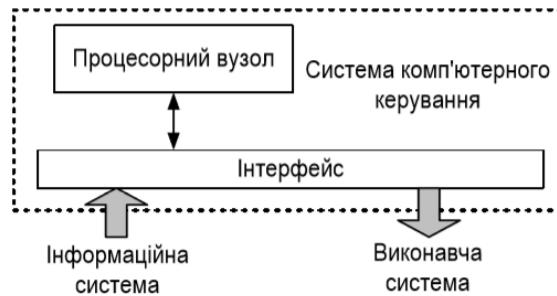


Рис. 6.2. Структура системи комп'ютерного керування з однопроцесорною організацією обчислювального пристрою

Перевагами такої системи комп'ютерного керування можна вважати простоту її організації й досить низьку вартість. Однак, з другого боку, така організація системи комп'ютерного керування має й ряд істотних недоліків, а саме:

- низька надійність, оскільки відмова процесорного вузла призводить до відмови всієї системи комп'ютерного керування (тобто до нездатності системи комп'ютерного керування виконувати задану функцію);
- низька швидкодія, оскільки реалізація алгоритму керування в такому разі буде відбуватися послідовним способом.

Названі переваги й недоліки однопроцесорної організації системи комп'ютерного керування мехатронного об'єкта визначають й основну сферу їхнього застосування, а саме у нескладних мехатронних об'єктах, які працюють в некритичних умовах експлуатації.

Коли йдеться про керування складним мехатронним об'єктом, швидкодія однопроцесорної системи комп'ютерного керування зазвичай виявляється недостатньою для керування в реальному часі. У такому разі використовують розподілений ієрархічний спосіб організації системи комп'ютерного керування. Узагальнену структуру системи комп'ютерного керування, у якій застосовано такий спосіб організації обчислювального пристрою, наведено на рис. 6.3.

Обчислювальний пристрій системи комп'ютерного керування будують у вигляді багаторівневої ієрархічної мережі процесорних вузлів, у якій кожному процесорному вузлу верхнього рівня

підпорядковується кілька процесорних вузлів нижнього рівня (на рис. 6.3 показано два рівні). Очевидно, що порівняно з однопроцесорною організацією така організація обчислювального пристрою системи комп'ютерного керування має ряд переваг:

- по-перше, завдяки «розпаралелюванню» процедур реалізації алгоритму керування між окремими процесорними вузлами досягається більш висока швидкодія;
- по-друге, підвищується надійність обчислювального пристрою, оскільки відмова деяких процесорних вузлів нижнього рівня може не призводити до катастрофічних наслідків для мехатронного об'єкта загалом.

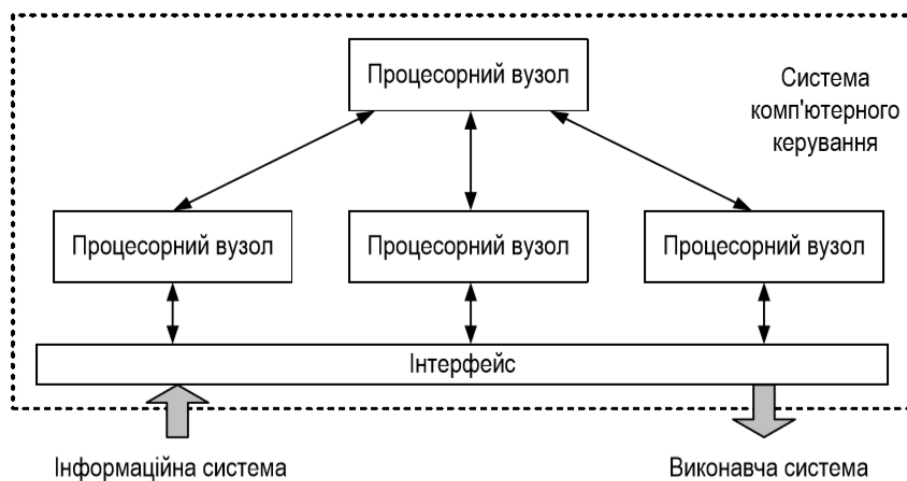


Рис. 6.3. Структура системи комп'ютерного керування з розподіленою ієрархічною організацією обчислювального пристрою

Недоліки такої організації обчислювального пристрою системи комп'ютерного керування:

- суттєво (не менш ніж у N разів, де N – кількість процесорних вузлів в системі) збільшується вартість ієрархічного розподіленого пристрою порівняно з однопроцесорною організацією обчислювального пристрою;
- в системі залишається «вузьке місце», а саме процесорний вузол верхнього рівня, відповідальний за організацію спільної роботи всіх процесорних вузлів нижнього рівня. Відмова процесорного вузла верхнього рівня приводить до виходу з ладу всього мехатронного об'єкта;

- в такій структурі взаємодія процесорних вузлів нижнього рівня один з одним може відбуватися тільки через процесорний вузол верхнього рівня, що, у свою чергу, може призводити до значних часових затримок під час обміну даними між ними і, як наслідок, до зниження швидкодії системи комп'ютерного керування загалом.

На відміну від розподіленої ієрархічної організації системи комп'ютерного керування за розподіленої децентралізованої організації системи комп'ютерного керування (рис. 6.4) немає будь-якої підпорядкованості одних процесорних вузлів іншим. Всі процесорні вузли, що входять в обчислювальний пристрій системи комп'ютерного керування, можуть бути рівні з погляду функціональних можливостей. Така організація системи комп'ютерного керування мехатронного об'єкта, яка в літературі називається колективом обчислювачів, відкриває перспективи досягнення цілого ряду переваг:

- по-перше, система матиме дуже високу надійність. Справді, оскільки всі процесорні вузли рівні, то вихід з ладу кожного з них не призводитиме до катастрофічних наслідків, а функції процесорного вузла, який вийшов з ладу, можуть бути перерозподілені між працездатними процесорними вузлами;
- по-друге, забезпечується можливість прямого обміну даними між будь-якими парами процесорних вузлів, а не через процесорний вузол верхнього рівня, як за ієрархічної організації, що суттєво знижує тимчасові втрати, натомість збільшує швидкість обчислювального пристрою.

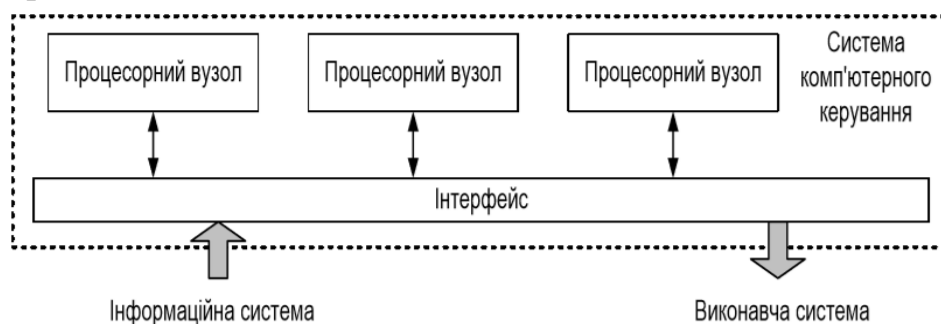


Рис. 6.4. Структура системи комп'ютерного керування з розподіленою децентралізованою організацією обчислювального пристрою

SCM (Supply Chain Management, управління ланцюгами поставок) – це управлінська концепція й організаційна стратегія, яка полягає в інтегрованому підході до управління всім потоком інформації (про сировину, матеріали, продукти, послуги), що виникає і перетворюється

під час виконання підприємством логістичних і виробничих процесів. Мета цієї стратегії – отримання вимірного сукупного економічного ефекту (зниження витрат, задоволення попиту на кінцеву продукцію).

До складу SCM-системи зазвичай входять дві підсистеми:

SCP (supply chain planning, планування ланцюжків постачань) – планування, формування календарних графіків, проєктування мереж поставок, моделювання різних ситуацій, аналіз виконання завдань;

SCE (supply chain execution, виконання ланцюжків постачань) – відстеження логістичних процедур і контроль їхнього виконання.

Поняття SCM логістики притаманне для процесів управління логістичними потоками вантажів у сфері доставки, проте це поняття також можна застосувати до системи управління процесами потоків інформації в мехатронній системі (рис. 6.5).

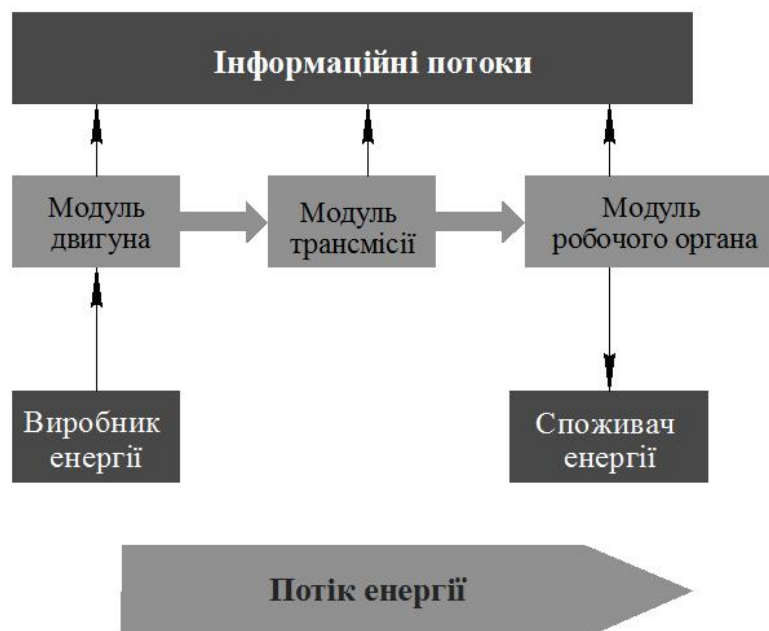


Рис. 6.5. Логістика інформаційних потоків в мехатронній системі

У процесі інтеграції компонентів мехатронної системи впроваджують нові інформаційні логістичні процеси на основі об'єднання ряду завдань та нової організації інформаційних потоків. Таким чином формується синергетичний підхід у побудові мехатронної системи. Важливим результатом інтеграції є створення єдиної системи із самокеруванням і самооптимізувальними ланками.

Збільшення кількості інформаційних мереж значно спрощує розвиток SCM-логістики в мехатронних системах. Для вдалого застосування логістичної інформації в мережі комунікації мехатронних

систем, слід створювати нові мережі комунікації й обробки інформації, орієнтовані на конкретні процеси мехатронної системи. Переконливим прикладом є створення так званого «інтернету речей» (IoT), де кожна складова такої систем повинна мати комунікацію з Інтернетом, а інформація з таких пристроїв обробляється на різних рівнях інформаційної системи – і на рівні самого пристрою, і на рівні системи управління.

На рис. 6.6 показано систему ключових компетенцій SCM логістики мехатронних систем.

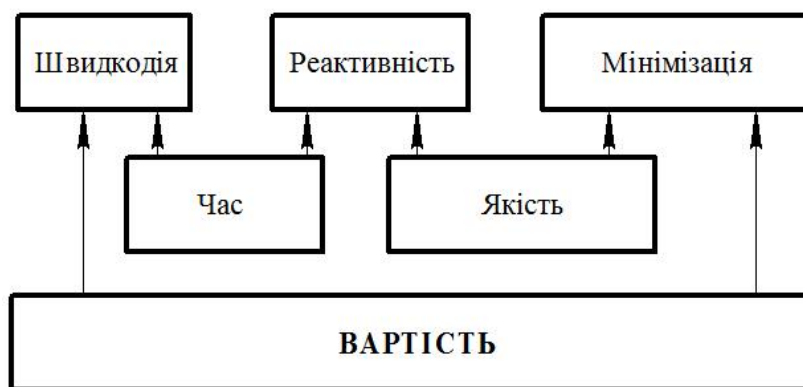


Рис. 6.6. Компетенції SCM-логістики

Промисловий «Інтернет речей» (Internet of Things, IoT) – це комп'ютеризація всіх робочих місць на підприємстві шляхом створення єдиної інформаційної мережі, в яку об'єднано всі виробничі об'єкти, зокрема й технологічне обладнання та робочі місця, завдяки цьому створюється екосистема, всі складові компоненти якої починають «розуміти» своє оточення і взаємодіяти між собою за інтернет-протоколом, самостійно вирішуючи питання підвищення ефективності або запобігання позаштатним ситуаціям.

Промислова автоматизація також починає досліджувати та впроваджувати концепцію і технології «Інтернет речей», розвинувши до концепцій: пристрій–пристрій (M2M), «Інтернет всього», «Інтернет речей краю».

Нині створено велику кількість «розумних» пристроїв, зокрема датчиків з виходом в мережу Internet, здатних передавати інформацію за IP-протоколом, забезпечуючи доставку пакетів даних від одного постачальника до іншого, орієнтуючись тільки на IP-адреси в заголовках пакетів. Такий підхід дає виробнику та клієнтам змогу віддалено

контролювати роботу виробничих майданчиків, своєчасно виконувати регламентні роботи, передбачати аварії і проводити плановий запобіжний ремонт або, наприклад, заздалегідь підготувати потрібні деталі на заміну, а знаючи фактичне і плановане завантаження виробничого обладнання, з'єданого з мережею, можна організувати автоматичну мережу замовлень між різними виробництвами, оптимізуючи таким чином логістичний ланцюг від постачальників до споживачів. Наприклад, обладнання заводу Philips в Нідерландах працює в неосвітленому приміщенні, де всю роботу виконують 128 роботів, а всі процеси заводу контролює персонал, який складається з дев'ятох працівників. В компаніях Harley Davidson, Volkswagen також застосовано підхід IoT, що дало змогу підвищити реакцію виробництва на запити споживачів в умовах зростання конкуренції та підвищити систему якості продукції.

Одним з прикладів ефективного впровадження технології IoT у промисловість є застосування в сучасних верстатах з числовим програмним керуванням (ЧПК) систем комп'ютеризованого управління. Наприклад програмна платформа IoT Winnum (рис. 6.7), основана на Java EE, є інтегрованим середовищем збору, зберігання й обробки великих обсягів даних, що надходять від різних мережних пристроїв, та виконує підтримку користувацьких застосунків для роботи з ними.

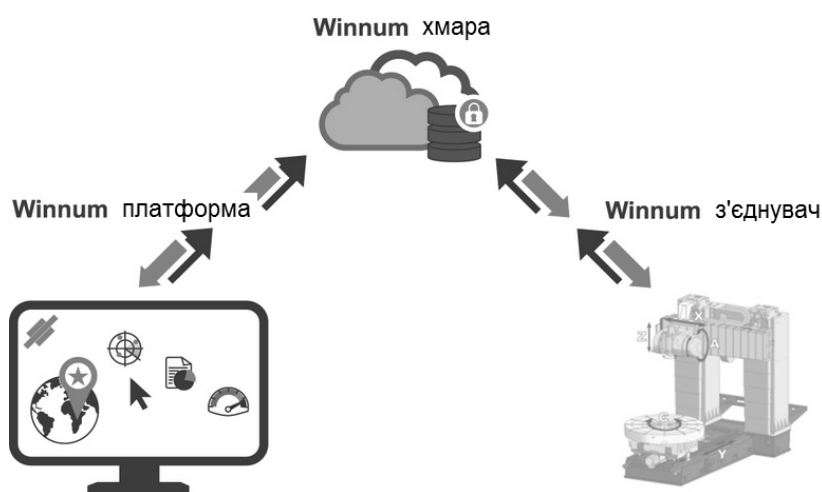


Рис. 6.7. Основні програмні компоненти Winnum

Компанія Fanuc пропонує систему комп'ютерного програмного забезпечення MT-LINKi, яка працює як локальний сервер обладнання для управління, збору, візуалізації даних з верстатів на підприємстві [2].

Система дає змогу під'єднувати не тільки верстати з ЧПК FANUC, а й інші периферійні пристрої (наприклад, промислові контролери ПЛК). MT-LINKi може бути основою для під'єднання обладнання до IoT, оскільки дає змогу під'єднати різні датчики і збирати з них інформацію (рис. 6.8) та передавати дані далі на сервери мережі.



Рис. 6.8. Комунікація на рівні системи MT-LINKi: OPC Unified Architecture (OPC) – специфікація, що визначає передавання даних в промислових мережах і взаємодію пристроїв в цих мережах

Інша система для контролю виробництва та управління продуктивністю верстатів – OMATIVE Pro, яку встановлюють на сервер заводської комп'ютерної мережі і з'єднують з системами керування верстатів. Програма відображає стан верстата: тривалість повного циклу обробки й окремих операцій, час перебування інструмента в матеріалі, динамічні зміни подавання і навантаження, програма також виконує статистичний збір даних.

Для стандартизації рішень щодо автоматизації виробництва на сучасних заводах застосовують CIM (computer-integrated manufacturing) підхід (рис. 6.9), згідно з яким комп'ютери повинні керувати всіма виробничими процесами: проектування, аналіз, планування, закупівлі, облік витрат, управління запасами і логістикою, що забезпечує прямий контроль і моніторинг всіх етапів життєвого циклу.

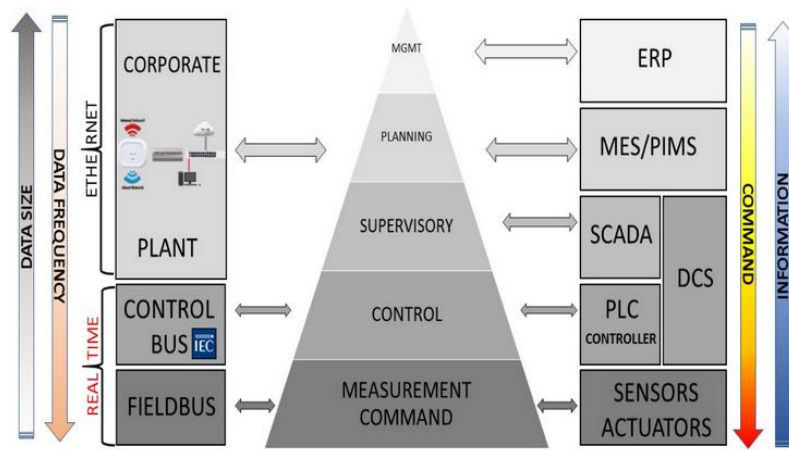


Рис. 6.9. Ієрархічні рівні системи комплексного автоматизованого виробництва СІМ: sensors, actuators – рівень датчиків, перетворювачів і виконавчих механізмів; PLC, DCS – рівень мікроконтролерів, розподілених систем керування або терміналів віддаленого управління мікропроцесорами (Remote terminal unit RTU); SCADA – рівень систем збору, обробки, відображення та архівування інформації про об'єкт моніторингу або управління в реальному часі через людино-машинний інтерфейс; MES – рівень синхронізації управління виробничої системи шляхом об'єднання рівнів планування і контролю для оптимізації процесів і ресурсів; ERP – рівень програмного забезпечення управління та планування діяльністю

Компанія Amazon – активний постачальник послуг з обслуговування пристроїв IoT на своїх сервісах AWS Cloud пропонує декілька сценаріїв використання промислового «Інтернету речей».

1. *Аналітика та прогнозування якості продукції і послуг компанії.* Згідно з цим сценарієм промислова компанія на своїх виробничих лініях повинна інтегрувати IoT у власне обладнання та під'єднатися до локального сервісу AWS IoT Greengrass, який розміщується на серверах підприємства. Цей сервіс дає змогу на основі моделей машинного навчання та фільтрації зібраних даних про пристрої виконувати їхнє програмування в локальній мережі та акумулювати і зберігати незначний об'єм даних (рис. 6.10), при цьому в ньому реалізовано окремо протоколи для обміну даними про OPC-UA, MQTT, HTTP, MODBUS.

HTTP (HyperText Transfer Protocol) – протокол передавання гіпертексту, тобто даних у вигляді символів рядка і відповідно до специфікації OSI є протоколом прикладного (верхнього, 7-го) рівня.

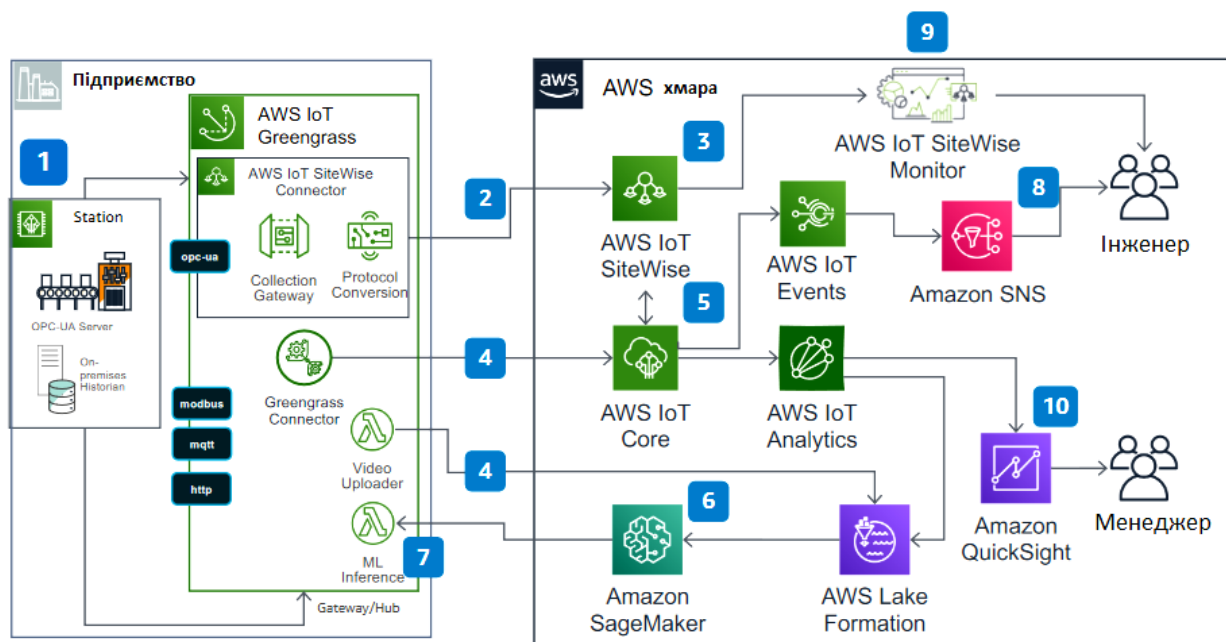


Рис. 6.10. Архітектура сервісів AWS Cloud для рішення аналітики та прогнозування: 1 – промислове обладнання з IoT; 2 – канал для під'єднання та збору даних із заводських машин за допомогою OPC-UA та комунікації між локальним AWS IoT SiteWise Connector та мережним AWS IoT SiteWise; 3 – сервіс мережної системи масштабування та зберігання даних AWS IoT SiteWise; 4 – головний канал під'єднання локального сервера AWS IoT Greengrass для обміну повідомленнями з глобальною мережею Amazon AWS IoT Core; 5 – сервіс обміну даними між AWS IoT Core та AWS IoT Event, який використовують для реагування на дані про дії IoT та аналітичні дані; 6 – сервіс прогнозування якості моделлю машинного навчання за допомогою Amazon SageMaker на основі зображень, що зберігаються в AWS Lake Formation; 7 – локальна система машинного навчання в шлюзі AWS IoT Greengrass Edge; 8 – канал сповіщень у службі Amazon Simple Notification Service (SNS); 9 – веб-портал для візуалізації даних в реальному часі; 10 – портал статистики проаналізованих даних

MQTT (Message queuing telemetry transport) – спрощений мережний протокол, який працює поверх TCP/IP, орієнтований на обмін повідомленнями між пристроями (за принципом «видавець–підписник» без обмеження на формат передавання даних і застосовуваний для роботи з датчиками та пристроями IoT.

Modbus – комунікаційний протокол, оснований на архітектурі «керівний–ведений». який використовує для передавання даних через стандарти RS-485, RS-422, RS-232, а також Ethernet мережі TCP/IP.

Таким чином, за допомогою описаної системи можна отримувати інформацію з промислового обладнання, навколишнього середовища і візуального спостереження (комп'ютерного зору (ML)).

2. *Архітектура системи моніторингу ресурсів* (рис. 6.11) дає змогу збирати дані про стан машин й устаткування та прогнозувати можливості використання ресурсу обладнання та заводу.

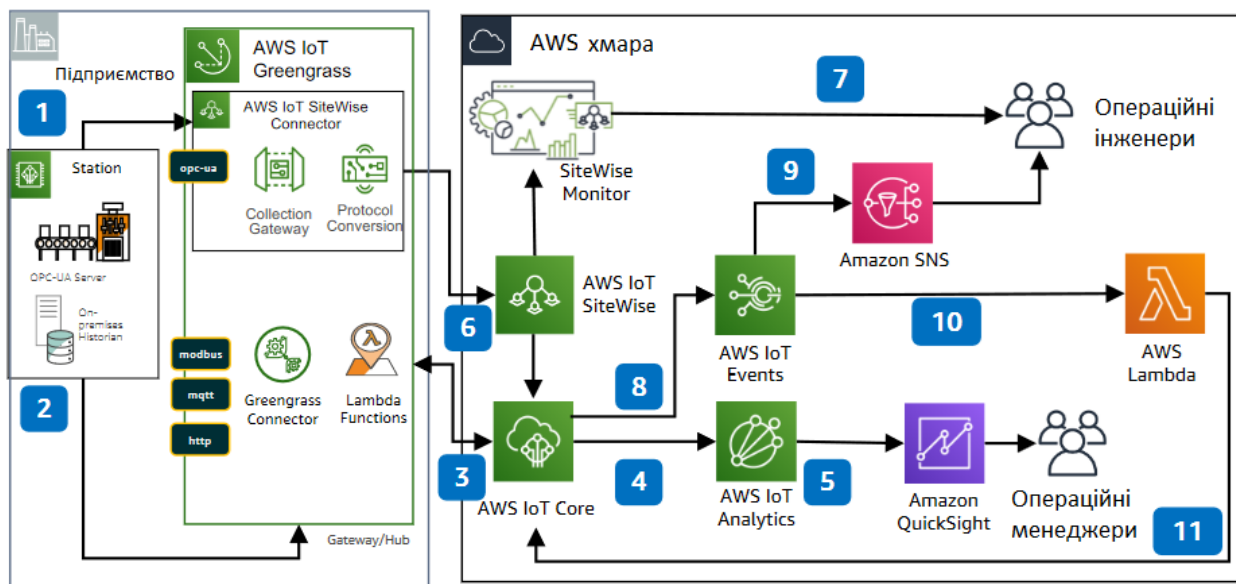


Рис. 6.11. Архітектура системи AWS Cloud моніторингу ресурсів: 1 – промислове обладнання з IoT; 2 – сервіс локальної обробки та аналізу даних на підприємстві; 3 – канал мережного ядра AWS IoT Core, що ініціює події, які надсилаються до AWS IoT Events та AWS IoT Analytics; 4 – канал передавання даних до мережних служб швидкого аналізу даних; 5 – передавання даних до сервісу масштабування, статистики й аналітики даних; 6 – сервіс моделей для збереження та відслідковування змін даних; 7 – веб-портал AWS IoT SiteWise Monitor для візуалізації даних у реальному часі; 8 – канал служби повідомлень та відслідковування подій; 9 – служба публікації та розсилання повідомлень на основі подій; 10 – сервіс мережних розрахунків; 11 – канал зв'язку між AWS IoT Core та AWS Lambda

3. *Архітектура система аналітики технічного обслуговування обладнання* (рис. 6.12) дає змогу збирати дані про стан промислового обладнання, виявляти потенційні несправності до того, як вони вплинуть на виробництво. Така система допомагає збільшити термін роботи обладнання, гарантувати безпечні умови праці робітників, оптимізувати логістичні процеси підприємства, безперервно відстежувати й аналізувати стан і продуктивність обладнання в реальному часі.

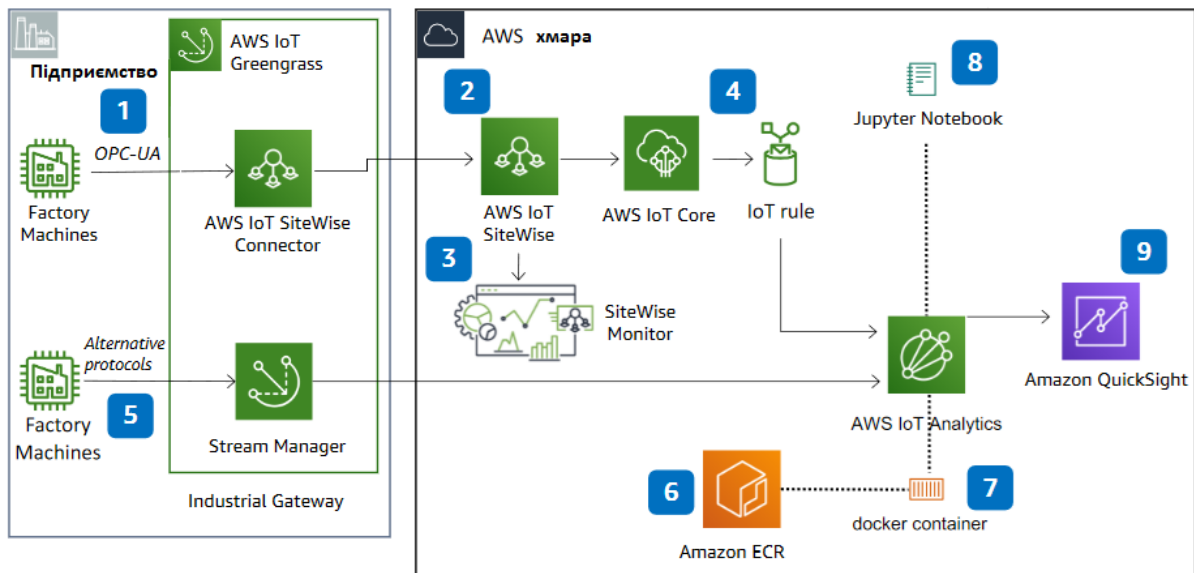


Рис. 6.12. Архітектура системи AWS Cloud аналітики технічного обслуговування: 1 – канал комунікації для під'єднання та збору даних із заводських машин за допомогою OPC-UA; 2 – сервіс моделей для збереження та відслідковування змін даних; 3 – веб-портал для візуалізації заводських даних та показників промислової продуктивності в режимі реального часу; 4 – сервіс резервного копіювання та передавання повідомлень до аналітики; 5 – менеджер потоків для паралельної та швидкої обробки даних; 6 – менеджер контейнерів стану системи для оброблених даних; 7 – Docker container; 8, 9 – додаткові інструменти для взаємодії з користувачами

Застосування таких рішень в промислових системах потребує реалізації під'єднання промислового обладнання до засобів комунікації з Internet-протоколами або з протоколами систем автоматики, як, наприклад, MQTT-протокол. В подальшому таке обладнання легко під'єднується до готових промислових захищених сервісів.

Контрольні запитання

1. Назвіть основні принципи SCM-логістики.
2. Що таке «Інтернет речей»?
3. Наведіть архітектуру інформаційних систем управління мехатронним модулем.

Лекція 7. Процеси логістики будівельної техніки

Для вибору і програмування мехатронної системи машини спочатку потрібно виконати аналіз руху її складових елементів загалом під час сталого руху (коли швидкості всіх частин механізму постійні). Крім того, визначають також можливий діапазон варіацій головних параметрів системи.

На початковому етапі аналізу процесів мехатронних систем застосовують статичний та кінематичний розрахунки з метою визначити сили та швидкості на окремих ланках мехатронної системи. Оскільки більшість механізмів мехатронних систем можна представити уніфікованими схемами – привід (двигун, гальмо), передача та робочий орган, тоді цю схему можна подати у такому вигляді: керівна ланка – передача – ведена ланка (рис. 7.1).



Рис. 7.1. Абстрактне подання механічної системи: Д – двигун;
П – привід; Р – робочий орган

Якщо індексом «1» позначити параметри керівної ланки, а індексом «2» –веденої, то залежності між цими параметрами можна виразити за допомогою таких рівностей:

$$M_{(2)} = M_{(1)} \cdot i_3 \cdot \eta_3; \quad (7.1)$$

$$N_2 = N_1 \cdot \eta_3; \quad (7.2)$$

$$\omega_2 = \frac{\omega_1}{i_3}; \quad (7.3)$$

$$n_2 = \frac{n_1}{i_3}, \quad (7.4)$$

де M – обертовий момент; i_3 , η_3 – відповідно загальне передатне число та загальний ККД; N – потужність; ω – кутова швидкість; n – кількість обертів.

Загальне передатне число між джерелом енергії і веденою ланкою

$$i_3 = i_1 \cdot i_2 \cdot i_3 \cdot \dots \cdot i_m, \quad (7.5)$$

а загальний ККД

$$\eta_3 = \eta_1 \cdot \eta_2 \cdot \eta_3 \cdot \dots \cdot \eta_m, \quad (7.6)$$

де m – кількість передач; 1, 2, 3 – порядковий номер окремої передачі.

За несталою руху, за періоди пуску та гальмування статичного та кінематичного розрахунків не досить. За умов несталості швидкостей елементів механізму такі величини, як тривалість періодів розгону та гальмування, шлях гальмування, перевантаження двигуна та передач можна визначити лише на підставі динамічних розрахунків, у яких відображають як несталість швидкості, так й інерційність мас, які беруть участь у русі.

Для динамічного розрахунку використовують спрощену розрахункову схему механізму, яка складається з двигуна, маховика та гальма. Інерційною характеристикою такого механізму є зведений момент інерції $I_{зв}$ – момент інерції маховика, кінетична енергія якого є зведеною для всіх елементів механізму.

Кінетична енергія маховика:

$$E = \frac{I_{зв} \omega_{дв}^2}{2}, \quad (7.7)$$

де $\omega_{дв}$ – кутова швидкість вала двигуна.

Кінетична енергія елементів механізму загалом:

$$E = E_{об} + E_{п} = \sum_{i=1}^n \frac{I_i \omega_i^2}{2\eta_i} + \sum_{j=1}^z \frac{m_j v_j^2}{2\eta_j}, \quad (7.8)$$

де $E_{об}$ та $E_{п}$ – кінетична енергія елементів, які рухаються відповідно обертально та поступально; $I_{зв}$, ω_i – момент інерції та кутова швидкість обертального i -го елемента; η_i – ККД передачі між i -м елементом та двигуном; n – кількість обертальних елементів; m_j , v_j – маса та швидкість j -го елемента, що рухається поступально; η_j – ККД передачі між j -м елементом та двигуном; z – кількість елементів, що рухаються поступально.

Із рівнянь (7.7) та (7.8) визначасмо зведений момент інерції механізму:

$$I_{зв} = \sum_{i=1}^n \frac{I_i \omega_i^2}{\omega_{дв}^2 \eta_i} + \sum_{j=1}^z \frac{m_j v_j^2}{\omega_{дв}^2 \eta_j}. \quad (7.9)$$

З цього виразу випливає, що зведений момент інерції обернено пропорційний до передатних чисел. Для машин безперервного транспорту (наприклад, для стрічкових конвеєрів) рівняння (7.9) може бути спрощене до такого вигляду:

$$I_{зв.к} = j(I_p + I_m) \frac{1}{\eta_{мех}} + \frac{m v^2}{\omega_d^2 \eta_{мех}}, \quad (7.10)$$

де $j=1,15...1,2$ – коефіцієнт обчислення моменту інерції оберткових деталей, які перебувають поза віссю вала двигуна; I_p і I_m – моменти інерції ротора двигуна і з'єднувальної муфти; m – маса вантажу разом з робочою і холостою гілками конвеєра, кг.

Величину m обчислюють за формулою

$$m = (q + 2q_1)L, \quad (7.11)$$

де q – погонна маса вантажу, кг/м; q_1 – погонна маса гілки конвеєра, кг/м; L – довжина конвеєра, м.

Тоді будемо мати з формули (7.11):

$$I_{зв.к} = j(I_p + I_m) \frac{1}{\eta_{мех}} + \frac{(q + 2q_1)L v^2}{\omega_d^2 \eta_{мех}}. \quad (7.12)$$

Для відповідної розрахункової схеми на рис. 7.1 загальне диференціальне рівняння руху механізму вантажопідйомної машини матиме такий вид:

$$M_{дв} - M_c = I_{зв} \frac{d\omega}{dt}, \quad (7.13)$$

де $M_{дв}$ і M_c – момент, відповідно, рушійних сил та сил статичного опору, зведений до вала двигуна; ω – кутова швидкість вала двигуна; t – час.

Ліва частина формули (7.13) є надлишковим, або динамічним, моментом, який спричиняє кутове прискорення $d\omega/dt$.

У загальному вигляді

$$M_{\text{дв}} = f(t), \quad (7.14)$$

причому функція $f(t)$ є різною для різних приводів. Як відомо, механізмам машин можна надати руху від асинхронних електродвигунів трифазного струму, від електродвигунів постійного струму (за системою генератор – двигун), від двигунів внутрішнього згоряння через фрикційну муфту, гідротрансформатор або гідравлічний пристрій, який вмикається золотником.

Для того щоби визначити тривалість періодів пуску та гальмування, а також швидкість та шлях у ці періоди, треба інтегрувати одержане рівняння руху. Для цього потрібно знати закони зміни прискорення або надлишкового моменту. Експериментальні дослідження приводів машин свідчать, що прискорення механізму з приводом від асинхронного електричного двигуна певною мірою стале, а швидкість зростає майже лінійно. У зв'язку з цим

$$\frac{d\omega}{dt} \approx \frac{\omega_{\text{дв}}}{t_{\text{п}}}, \quad (7.15)$$

де $t_{\text{п}}$ – тривалість пуску.

Із формули (7.15) та рівняння (7.13) знаходимо

$$t_{\text{п}} = \frac{I_{\text{зв}} \omega_{\text{дв}}}{M_{\text{дв}} - M_{\text{с}}}. \quad (7.16)$$

Рухійний момент на валу двигуна у процесі пуску механізму (пусковий момент) є змінною величиною. Він збільшується від нуля до максимального значення $M_{\text{дв}}^{\text{max}}$, яке є характеристикою двигуна, що наводиться у каталогах, а потім падає до величини, незмінної за сталого руху. Для розрахунків користуються середнім значенням пускового моменту:

$$M_{\text{пс}} = \psi \cdot M_{\text{н}}, \quad (7.17)$$

де ψ – кратність середнього пускового моменту двигуна ($\psi = 1,5 \dots 1,6$ – для електродвигунів трифазного струму з фазним ротором, а для двигунів постійного струму зі збудженням: паралельним – $\psi = 1,7 \dots 1,8$; послідовним – $\psi = 1,8 \dots 2,0$; змішаним – $\psi = 1,8 \dots 1,9$); $M_{\text{н}}$ – номінальний момент двигуна:

$$M_{\text{н}} = \frac{N}{\omega_{\text{дв}}}, \quad (7.18)$$

N – номінальна потужність двигуна.

Для двигунів трифазного струму з короткозамкненим ротором:

$$M_{\text{пс}} = (0,7 \dots 0,8) M_{\text{дв}}^{\text{max}}. \quad (7.19)$$

Тоді тривалість пуску механізму:

$$t_{\text{п}} = \frac{I_{\text{зв}} \omega_{\text{дв}}}{M_{\text{пс}} - M_{\text{с}}}. \quad (7.20)$$

Під час гальмування надлишковий момент дорівнює сумі $M_{\text{Г}} + M_{\text{с}}$, де $M_{\text{Г}}$ – гальмівний момент гальма, зведений до вала двигуна. Аналогічно до виразу (7.13) тривалість гальмування механізму становитиме:

$$t_{\text{Г}} = \frac{I'_{\text{зв}} \omega_{\text{дв}}}{M_{\text{Г}} + M_{\text{с}}}, \quad (7.21)$$

де $I'_{\text{зв}}$ – зведений до вала двигуна момент інерції механізму під час гальмування, який визначають, зважаючи на те, що тертя сприяє гальмуванню:

$$I'_{\text{зв}} = \sum_{i=1}^n \frac{I_i \omega_i^2 \eta_i}{\omega_{\text{дв}}^2} + \sum_{j=1}^z \frac{m_j v_j^2 \eta_j}{\omega_{\text{дв}}^2}. \quad (7.22)$$

Середнє кутове прискорення вала двигуна механізму:

- під час пуску

$$\varepsilon_{\text{п}} = \frac{\omega_{\text{дв}}}{t_{\text{п}}}; \quad (7.23)$$

- під час гальмування

$$\varepsilon_{\text{Г}} = \frac{\omega_{\text{дв}}}{t_{\text{Г}}}. \quad (7.24)$$

Кут повороту вала двигуна

- під час пуску

$$\varphi_{\Pi} = \frac{\omega_{\text{дв}} t_{\Pi}}{2}; \quad (7.25)$$

- під час гальмування

$$\varphi_{\Gamma} = \frac{\omega_{\text{дв}} t_{\Gamma}}{2}. \quad (7.26)$$

Потрібний момент M_c (Н·м) на валу двигуна для подолання статичного опору можна визначати за формулою

$$M_c = \frac{M_{\text{пр.л}}}{u_{\text{заг}} \eta_{\text{мех}}} = \frac{F_{\text{роз}} D_{\text{пр}}}{2u_{\text{заг}} \eta_{\text{мех}}} = \frac{30F_{\text{роз}} \nu}{\pi n_{\text{пр}} u_{\text{заг}} \eta_{\text{мех}}} = \frac{9,55F_{\text{роз}} \nu}{n_{\text{д}} \eta_{\text{мех}}}, \quad (7.27)$$

де $M_{\text{пр.л}}$ – крутний момент на керівний ланці, Н·м; $n_{\text{д}}$ – частота обертання вала двигуна, хв.⁻¹; $n_{\text{пр}}$ – частота обертання тягового вала (барабана чи зірочки), хв.⁻¹; $u_{\text{заг}}$ – загальне передатне відношення привода; $D_{\text{пр}}$ – діаметр головної ланки конвеєра (барабана чи зірочки), м.

Підставляючи залежність у формули (7.12) і (7.27), визначимо тривалість пуску транспортувальної машини з огляду на особливості її будови:

$$t_{\Pi} = \frac{j(I_p + I_m) \frac{\omega_{\text{д}}}{\eta_{\text{мех}}} + \frac{(q + 2q_1)L \nu^2}{\omega_{\text{д}} \eta_{\text{мех}}}}{M_{\text{пс}} - \frac{P_{\text{роз}} \nu}{\omega_{\text{д}} \eta_{\text{мех}}}}. \quad (7.28)$$

Рекомендовано, що час розгону асинхронних двигунів не повинен перевищувати 5 с. Якщо за формулою (7.27) час розгону буде більшим, то слід збільшити потужність двигуна і виконати повторний розрахунок.

Розглянемо деякі особливості розрахунку загальних параметрів мехатронного приводу будівельної техніки.

Гвинтовий привід (рис. 7.2).

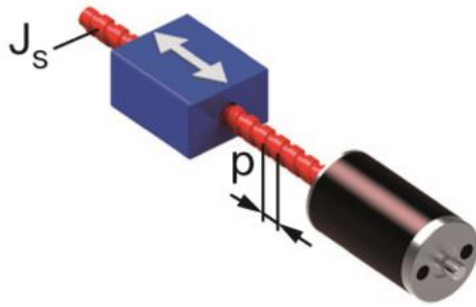


Рис. 7.2. Розрахункова схема гвинтового приводу

Частота обертання гвинта, об/хв:

$$n_{in} = \frac{60}{p} v_L, \quad (7.29)$$

де p – гвинтовий крок, м; v_L – швидкість навантаження, м/с.

Статичний крутний момент на валу двигуна, Н·м::

$$M_{in} = \frac{p}{2\pi} \frac{F_L}{\eta}, \quad (7.30)$$

де F_L – зовнішня сила навантаження, Н; η – ККД передачі.

Додатковий крутний момент для постійного прискорення, Н·м:

$$M_{in_a} = (J_{in} + J_s + \frac{m_L + m_s}{\eta} \frac{p^2}{4\pi^2}) \frac{\pi}{30} \frac{\Delta n_{in}}{\Delta t_a}, \quad (7.31)$$

де J_{in} , J_s – моменти інерції на вході (двигун, кодер, гальмо) та гвинта, відповідно, кгм²; Δt_a – період, протягом якого відбувається зміна частоти обертання Δn_{in} ; m_L , m_s – маса вантажу та гайки, кг.

Похибка позиціювання каретки:

$$\Delta \varphi_{in} = \Delta s_L \frac{2\pi}{p}. \quad (7.32)$$

Барабанний привід стрічкового конвеєра та лебідки крана (рис. 7.3 та 7.4)

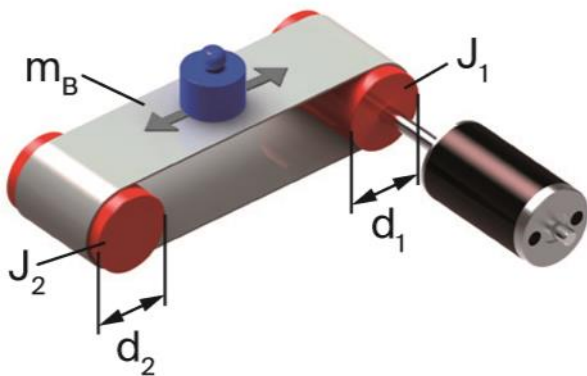


Рис. 7.3. Розрахункова схема приводу стрічкового конвеєра

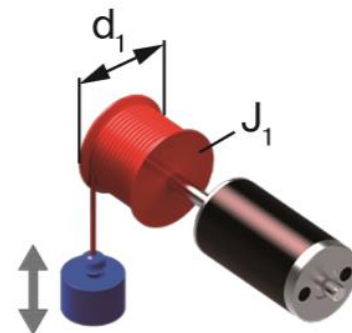


Рис. 7.4. Розрахункова схема канатного приводу барабана

Частота обертання приводного барабана, якщо немає проковзування:

$$n_{in} = \frac{60}{\pi} \frac{v_L}{d_1}, \quad (7.33)$$

де d_1 – діаметр приводного барабана, м.

Статичний крутний момент двигуна, Н·м:

$$M_{in} = \frac{d_1}{2} \frac{F_L}{\eta}, \quad (7.34)$$

де F_L – зовнішня сила навантаження, Н; η – ККД передачі.

Додатковий крутний момент для постійного прискорення, Н·м:

$$M_{in_a} = (J_{in} + J_1 + \frac{J_2}{\eta} \frac{d_1^2}{d_2^2} + \frac{m_L + m_B}{\eta} \frac{d_1^2}{4}) \frac{\pi}{30} \frac{\Delta n_{in}}{\Delta t_a}, \quad (7.35)$$

де J_1, J_2 – момент інерції веденого та головного шківів, кгм²; m_L, m_B – маса стрічки та вантажу, кг; Δt_a – період, протягом якого відбувається зміна частоти обертання барабана Δn_{in} .

Похибка позиціювання каретки:

$$\Delta \varphi_{in} = \Delta s_L \frac{2}{d_1}. \quad (7.36)$$

Рейковий привід (рис. 7.5)

Частота обертання вала двигуна, об/хв:

$$n_{in} = \frac{60}{pz} v_L, \quad (7.37)$$

де p – крок рейки; z – кількість зубців зубчастого колеса.

Статичний крутний момент двигуна, Н·м:

$$M_{in} = \frac{pz}{2\pi} \frac{F_L}{\eta}. \quad (7.38)$$

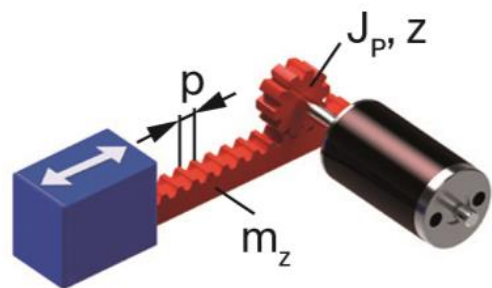


Рис. 7.5. Рейковий привід

Додатковий крутний момент для постійного прискорення:

$$M_{in_a} = (J_{in} + J_p + \frac{m_L + m_z}{\eta} \frac{p^2 z^2}{4\pi^2}) \frac{\pi}{30} \frac{\Delta n_{in}}{\Delta t_a}, \quad (7.39)$$

де m_L, m_z – маса зовнішнього вантажу та зубчастої рейки, кг.

Похибка позиціювання каретки:

$$\Delta\varphi_{in} = \Delta s_L \frac{2}{pz}, \quad (7.40)$$

де Δs_L – механічний люфт, навантаження, м.

Привід колісної каретки (рис. 7.6)

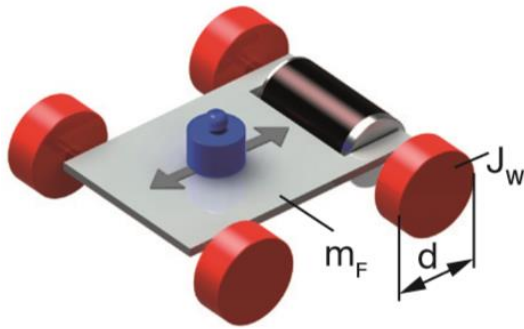


Рис. 7.6. Схема приводу колеса

Частота обертання колеса без проковзування:

$$n_{in} = \frac{60}{\pi} \frac{v_L}{d}, \quad (7.41)$$

де v_L – задана швидкість переміщення; d – діаметр колеса, м.

Статичний крутний момент двигуна, Н·м:

$$M_{in} = \frac{d}{2} \frac{F_L}{\eta}. \quad (7.42)$$

Додатковий крутний момент для постійного прискорення, Н·м:

$$M_{in_a} = (J_{in} + J_W + \frac{m_L + m_F}{\eta} \frac{d^2}{4}) \frac{\pi}{30} \frac{\Delta n_{in}}{\Delta t_a}, \quad (7.43)$$

де J_W – момент інерції всіх колес разом, кг·м²; m_L , m_F – маси каретки та вантажу, кг.

Похибка позиціювання каретки:

$$\Delta\varphi_{in} = \Delta s_L \frac{2}{d}. \quad (7.44)$$

Ексцентриковий привід (7.7)

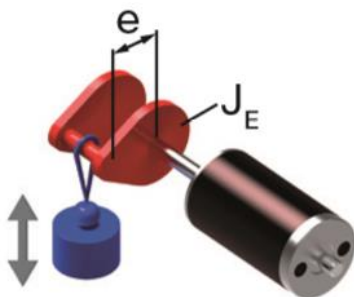


Рис. 7.7. Схема ексцентрикового приводу

Синусоїдальна крива швидкості навантаження за постійної вхідної швидкості, м/с:

$$v_L(t) = \frac{\pi}{30} n_{in} \cdot e \cdot \sin\left(\frac{\pi}{30} \cdot n_{in} \cdot t\right), \quad (7.45)$$

де e – ексцентриситет, м; t – період часу, с.

Зміна колового зусилля залежно від кута повороту ексцентрика, Н:

$$F_a(\varphi) = F_a \cdot \cos(\varphi) = m_L \cdot \left(\frac{\pi}{30} n_{in}\right)^2 \cdot e \cdot \cos \varphi, \quad (7.46)$$

де φ – кут повороту, радіан; F_a – сила прискорення, Н·м.

Залежність зміни крутного моменту від кута повороту для різних умов навантаження у двох напівциклах руху вперед і назад, Н·м:

$$M_{in1}(\varphi) = e \cdot (F_{L1} \cdot \sin \varphi + F_{a1} \cdot \cos \varphi). \quad (7.47)$$

Середній ефективний момент навантаження, Н:

$$M_{in1_RMS} = \frac{e}{\eta \sqrt{2}} \cdot \sqrt{F_{L1}^2 + F_{a1}^2 + F_{L2}^2 + F_{a2}^2}. \quad (7.48)$$

Додатковий крутний момент для прискорення ексцентрика, Н·м:

$$M_{in1_a} = (J_{in} + J_E) \cdot \frac{\pi}{30} \frac{\Delta n_{in}}{\Delta t_a}. \quad (7.49)$$

Механічна зубчаста передача (рис. 7.8)

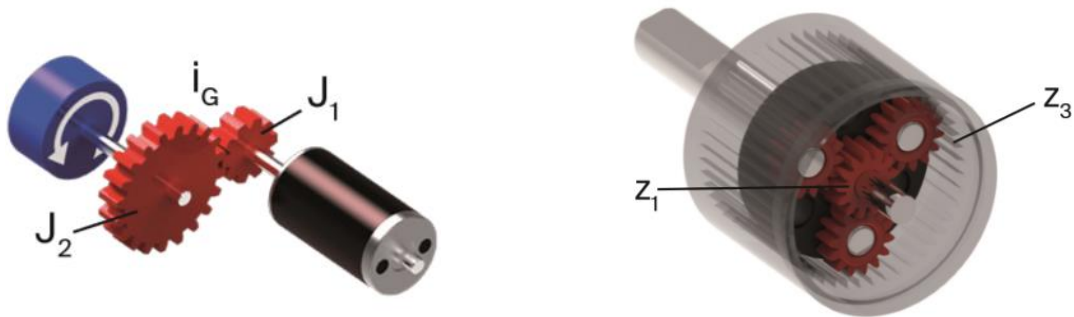


Рис. 7.8. Схема приводу механічної зубчастої передачі

Частота обертання колеса без проковзування, об/хв:

$$n_{in} = n_L i_G, \quad (7.50)$$

Середній ефективний момент навантаження, Н·м:

$$M_{in} = \frac{M_L}{i_G \eta}, \quad (7.51)$$

де M_L – момент на вихідному валу.

Додатковий крутний момент для постійного прискорення, Н·м:

$$M_{in_a} = \left(J_{in} + J_1 + \frac{J_L + J_2}{i_G^2 \eta}\right) \frac{\pi}{30} \frac{\Delta n_{in}}{\Delta t_a} = \left(J_{in} + J_G + \frac{J_L}{i_G^2 \eta}\right) \frac{\pi}{30} \frac{\Delta n_{in}}{\Delta t_a}, \quad (7.52)$$

де J_L – момент інерції на вихідному валу.

Похибка позиціювання:

$$\Delta\varphi_{in} = \Delta\varphi_L i_G. \quad (7.53)$$

Передатне число планетарного редуктора

$$i_G = \frac{z_1 + z_3}{z_1}, \quad (7.54)$$

Пасова передача (рис. 7.9)

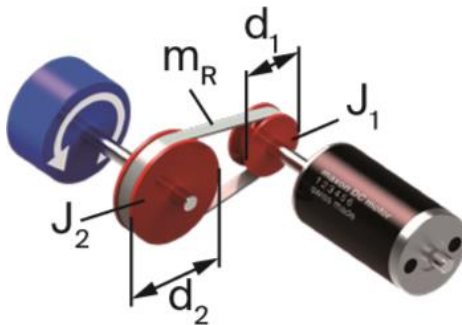


Рис. 7.9. Схема приводу механічної пасової передачі

Частота обертання колеса без проковзування, об/хв:

$$n_{in} = n_L \frac{d_2}{d_1}. \quad (7.55)$$

Середній ефективний момент навантаження:

$$M_{in} = \frac{d_1}{d_2} \frac{M_L}{\eta}, \quad (7.56)$$

де M_L – момент на вихідному валу.

Додатковий крутний момент для постійного прискорення:

$$M_{in_a} = (J_{in} + J_1 + \frac{J_L + J_2}{\eta} \frac{d_1^2}{d_2^2} + \frac{m_R d_1^2}{4\eta}) \frac{\pi}{30} \frac{\Delta n_{in}}{\Delta t_a}. \quad (7.57)$$

Похибка позиціювання:

$$\Delta\varphi_{in} = \Delta\varphi_{out} \frac{d_2}{d_1}. \quad (7.58)$$

Контрольні запитання

1. Наведіть основні характеристики механічних передач.
2. Що таке передаткова функція?
3. Як визначити крутний момент двигуна мехатронної системи приводу конвеєра?
4. Як визначити крутний момент приводу механічної передачі?
5. Які параметри передачі застосовують для керування її рухом?

Лекція 8. Логістика мехатроні системи

У теорії керування одне з ключових місць належить розробці та аналізу процесів автоматичного керування. Найпоширенішими проблемами автоматичного керування є задачі з побудови систем досягнення цілі та побудови або застосування автоматичних контролерів й автоматичних регуляторів. Всі ці задачі ґрунтуються на дослідженнях динаміки систем і математичного моделювання, теорії автоматичного керування, принципах математичної та інформаційної техніки.

Регулятори – це пристрої для керування параметрами стану системи (швидкості, прискорення, тиску, сили струму). Регулятори можуть використовуватися для обмеження параметрів на відповідних рівнях, наприклад, для повільної зміни стану параметра. Регулятори поділяються на дві групи: аналогові і цифрові. Аналогові регулятори використовують аналогові значення, такі як потік, тиск, температура і об'єм, які змінюються плавно. Цифрові регулятори використовують цифрові значення для керування параметрами вихідного значення і можуть задавати величини параметрів за допомогою цифрових значень, а потім перетворювати ці значення на аналогові.

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

Серед усієї різноманітності структур регуляторів варто виділити окремий тип – пропорційно-інтегрально-диференціальний регулятор (ПІД-регулятор). ПІД-регулятор призначений для виконання завдань з керування об'єктами з одним входом й одним виходом. Прикладом може бути сервопривід: завдання полягає в переміщенні механізму на задану відстань, якщо керівним параметром є швидкість обертання двигуна. Формалізація ПІД-регуляторів в загальному вигляді визначається такою математичною формулою:

$$u(t) = k_p e(t) + k_I \int_0^t e(t) dt + k_D \frac{d}{dt} e(t), \quad (8.1)$$

де $k_p > 0$, $k_I > 0$, $k_D > 0$ – деякі постійні коефіцієнти, вибір яких визначає поведінку регулятора (ці коефіцієнти називають часто пропорційним, інтегральним та диференціальним коефіцієнтами ПД-регулятора відповідно); $e(t) = r(t) - y(t)$ – похибка керування регулятора; $u(t)$ – керівний вплив; $r(t)$ – вхідний сигнал об'єкта керування (завдання, настанова); $y(t)$ – вихідний сигнал об'єкта керування (поточне або реально виміряне значення регульованої величини).

З формули (8.1) маємо структуру системи управління для об'єкта керування з ПД-регулятором і зворотним зв'язком (замкнена система керування) (рис. 8.1).



Рис. 8.1. Структура системи із застосуванням ПД-регулятора

Найпростішим способом визначення параметрів ПД-регулятора може бути певна стратегія ручного налаштування. Спочатку треба обнулити всі його коефіцієнти ($k_p=0$, $k_D=0$, $k_I=0$), що рівнозначно від'єднанню ПД-регулятора від об'єкта керування. Після цього починають працювати з типовим впливом – сходиною: на вхід $r(t)$ у якийсь момент часу подається постійний сигнал заданої амплітуди $r(t)=\text{const}$. Задача налаштування регулятора полягає у тому, щоб вихідна величина $y(t)$ об'єкта керування якнайкраще повторювала вхідний вплив $r(t)$. Для цього поступово збільшують k_p , постійно перевіряючи реакцію на вплив $r(t)=\text{const}$ доти, доки вихідна величина $y(t)$ не почне коливатися – це критичне значення коефіцієнта $k_p^{\max}$. Після цього потрібно встановити остаточне значення коефіцієнта $k_p = 0,5 k_p^{\max}$. Якість регулювання можна покращити, налаштовуючи коефіцієнти k_D та k_I . Якщо залишкова похибка занадто велика, тоді слід незначно збільшити k_I , якщо сильне перерегулювання – збільшити k_D .

Налаштування ПД-регулятора ґрунтується на певних правилах, які зводяться до такого:

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

У побудові регуляторів і контролерів одним з напрямів їхнього синтезу є розробка математичної моделі об'єкта управління. Математична модель динаміки роботи об'єкта управління потрібна для визначення характеристики реального стану системи та подальшого створення критеріїв управління привідним механізмом.

Розглянемо приклад побудови математичної моделі роботи механізму зміни вильоту вантажу стрілового крана з вантажною кареткою (рис. 8.2).

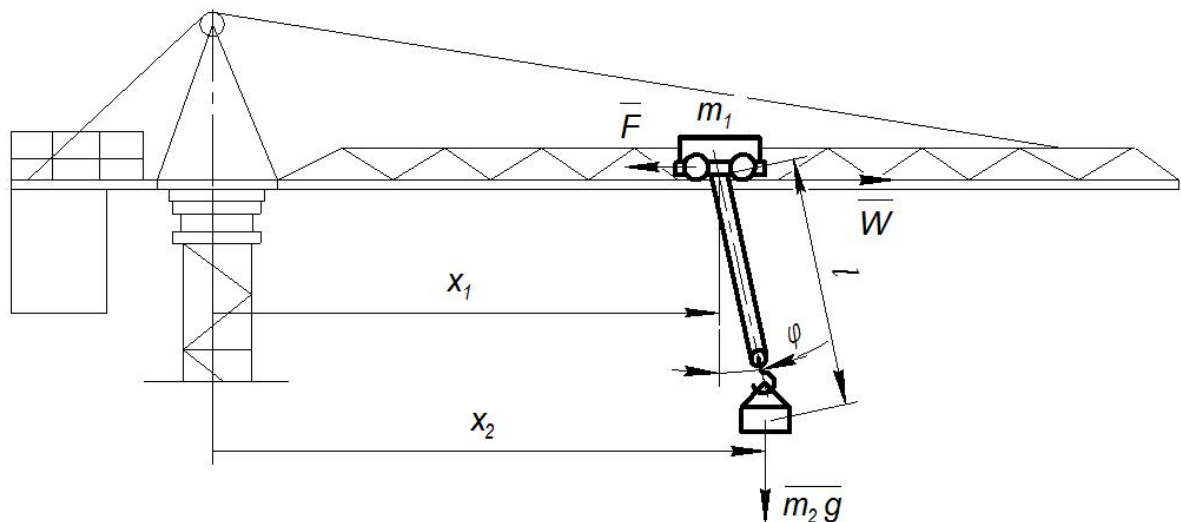


Рис. 8.2. Динамічна модель механізму зміни вильоту вантажу баштового крана з горизонтальною стрілою

Припускаємо, що всі елементи механізму зміни вильоту вантажу є твердими тілами, всі елементи крана рухаються у вертикальній площині, а вантаж коливається на гнучкому підвісі. Статичний опір переміщенню кранового візка є постійною величиною, оскільки відхилення поліспавної системи від вертикалі в процесі зміни вильоту вантажу є незначними і вони практично не змінюють величини статичного опору. Однак інерційні навантаження, що діють на візок унаслідок відхилення вантажу, є значними, і їх треба брати до уваги.

На основі прийнятих припущень запишемо рівняння, які визначатимуть динамічну модель руху механізму зміни вильоту вантажу:

$$\begin{cases} m_1 \ddot{x}_1 = F(t) - W \operatorname{sign} \dot{x}_1 - \frac{m_2 g}{l} (x_1 - x_2); \\ m_2 \ddot{x}_2 = -\frac{m_2 g}{l} (x_1 - x_2), \end{cases} \quad (8.2)$$

де m_1, m_2 – маси візка та вантажу, відповідно; $\ddot{x}_1, \ddot{x}_2, x_1, x_2$ – прискорення і переміщення візка та вантажу, відповідно; $F(t)$ – рушійна сила; W – сила опору переміщення візка; l – довжина підвісу вантажу.

Введемо додаткові змінні в систему рівнянь (8.2), зокрема:

$$\begin{aligned} y_1 &= x_1; \\ y_2 &= \dot{x}_1; \\ y_3 &= x_2; \\ y_4 &= \dot{x}_2. \end{aligned} \quad (8.3)$$

Беручи до уваги введені змінні (8.3), запишемо рівняння (8.2) в канонічній формі:

$$\begin{cases} \dot{y}_1 = y_2; \\ \dot{y}_2 = u(t) - \frac{m_2 g}{m_1 l} (y_1 - y_3); \\ \dot{y}_3 = y_4; \\ \dot{y}_4 = -\frac{g}{l} (y_1 - y_2), \end{cases} \quad (8.4)$$

де $u(t) = (F(t) - W \operatorname{sign}(y_2)) / m_1$ – функція керування системою.

Систему лінійних неоднорідних диференціальних рівнянь (8.3) можна звести до одного диференціального рівняння четвертого порядку:

$$\frac{d^4 y_3}{dt^4} + \lambda^2 \frac{d^2 y_3}{dt^2} = \frac{u(t)g}{l}, \quad (8.5)$$

$$\text{де } \lambda = \sqrt{\left(1 + \frac{m_2}{m_1}\right) \frac{g}{l}}; \quad \frac{d^2 y_3}{dt^2} = \ddot{y}_3; \quad \frac{d^4 y_3}{dt^4} = y_3^{IV}.$$

Застосовуючи метод Коші, можна знайти розв'язок рівняння (8.5), якщо відома фундаментальна система розв'язків однорідного рівняння, для якої слід розв'язати характеристичне рівняння:

$$k^4 + \lambda^2 k^2 = 0. \quad (8.6)$$

Розв'язок системи (8.4) запишемо у вигляді такої системи рівнянь:

$$\begin{aligned}
 y_1 &= b_1 + b_2 t - b_3 \frac{m_2}{m_1} \cos \lambda t - b_4 \frac{m_2}{m_1} \sin \lambda t + \\
 &+ \int_0^t \left[\frac{g}{l \lambda^2} (t - \tau) + \left(\frac{1}{\lambda} - \frac{g}{l \lambda^3} \right) \sin \lambda (t - \tau) \right] u(\tau) d\tau; \\
 y_2 &= b_2 + b_3 \lambda \frac{m_2}{m_1} \sin \lambda t - b_4 \lambda \frac{m_2}{m_1} \cos \lambda t + \\
 &+ \int_0^t \left[\frac{g}{l \lambda^2} + \left(1 - \frac{g}{l \lambda^2} \right) \cos \lambda (t - \tau) \right] u(\tau) d\tau; \\
 y_3 &= b_1 + b_2 t + b_3 \cos \lambda t + b_4 \sin \lambda t + \\
 &+ \int_0^t \left[\frac{g}{l \lambda^2} (t - \tau) - \frac{g}{l \lambda^2} \sin \lambda (t - \tau) \right] u(\tau) d\tau; \\
 y_4 &= b_2 - b_3 \lambda \sin \lambda t + b_4 \cos \lambda t + \\
 &+ \int_0^t \left[\frac{g}{l \lambda^2} - \frac{g}{l \lambda^2} \cos \lambda (t - \tau) \right] u(\tau) d\tau,
 \end{aligned} \tag{8.7}$$

де

$$\begin{aligned}
 b_1 &= (m_1 y_{10} + m_2 y_{30}) / m; \\
 b_2 &= (m_1 y_{20} + m_2 y_{40}) / m; \\
 b_3 &= (y_{30} - y_{10}) m_1 / m; \\
 b_4 &= (y_{40} - y_{20}) \lambda m_1 / m; \\
 m &= m_1 + m_2,
 \end{aligned}$$

де $y_{10}, y_{20}, y_{30}, y_{40}$ – початкові значення координат y_1, y_2, y_3, y_4 в момент часу $t = 0$.

Розглянемо математичну модель механізму вертикального підймання вантажу канатною лебідкою крана (рис. 8.3). В цій системі діє сила ваги вантажу $G = m_2 g$. Вважатимемо, що канат має пружність, а коефіцієнт пружності каната дорівнює c . Відомо, що момент інерції барабана лебідки становить J_1 , а маса вантажу, що підіймається, дорівнює m_2 .

Під дією зовнішніх сил в несталий період руху в пружній системі каната виникатимуть коливальні процеси, які можна описати системою диференціальних рівнянь:

$$\begin{cases} J_1 \ddot{\varphi} = M(t) - c(x_2 - x_1) \frac{D}{2}; \\ m_2 \ddot{x}_2 = c(x_2 - x_1) - m_2 g, \end{cases} \quad (8.8)$$

де x_2 , x_1 – переміщення вантажу та його початкове положення, відповідно; D – діаметр барабана; $M(t)$ – привідний момент на барабані; φ – кут повороту вала барабана.

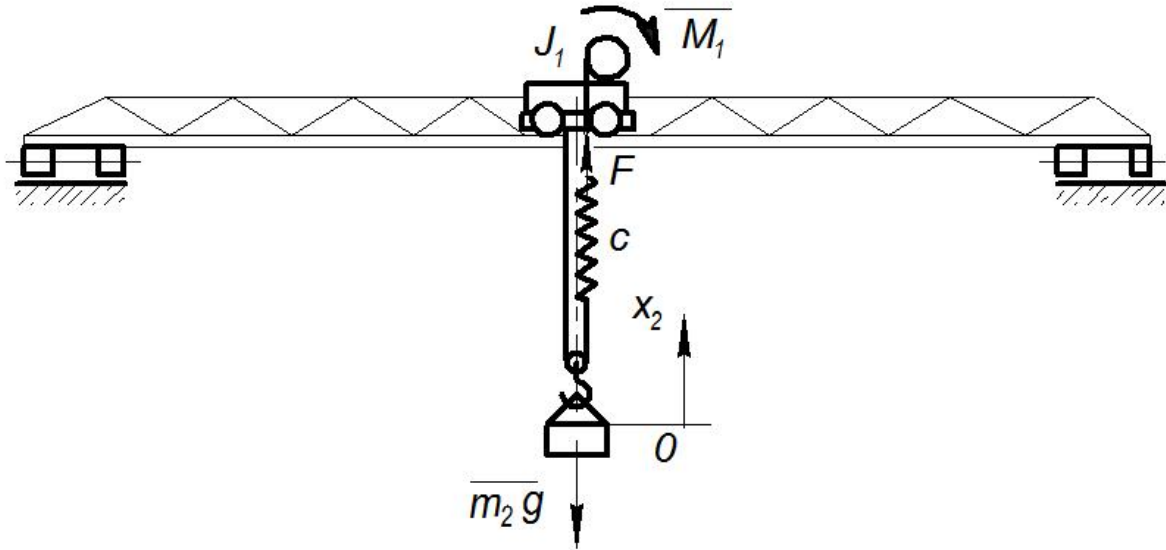


Рис. 8.3. Динамічна модель механізму зміни вильоту вантажу баштового крана з горизонтальною стрілою

Замінімо кутову координату φ на лінійну x_1 , тоді:

$$d\varphi = \frac{2}{D} dx_1; \quad (8.9)$$

$$\frac{d\varphi}{dt} = \dot{\varphi} = \frac{2}{D} \dot{x}_1; \quad (8.10)$$

$$\ddot{\varphi} = \frac{2}{D} \ddot{x}_1. \quad (8.11)$$

Запишемо в рівнянні (8.8) кутове прискорення через лінійне та домножимо всі складові першого рівняння системи (8.8) на D/J_1 , а другого – на $1/m_2$. Віднімемо від першого рівняння друге і в результаті отримаємо таке диференціальне рівняння:

$$\ddot{x} + cx \left(\frac{D^2}{4J_1} + \frac{1}{m_2} \right) = \frac{M(t)D}{2J_1} + g, \quad (8.12)$$

де $x = x_1 - x_2$.

Розв'язавши рівняння (8.12), отримаємо залежність для визначення динамічних зусиль в пружному зв'язку каната:

$$x = A \cos vt + B \sin vt + C, \quad (8.12)$$

де A, B – постійні інтегрування або амплітуди коливань динамічних навантажень; $C = \frac{2m_2 M(t)D + 4m_2 J_1 g}{c(m_2 D^2 + 4J_1)}$; v – власна частота коливань. Для двомасової системи:

$$v = \sqrt{c \left(\frac{m_2 D^2 + 4J_1}{4J_1 m_2} \right)}. \quad (8.13)$$

Період власних коливань:

$$T = \frac{2\pi}{v}. \quad (8.14)$$

У виробничому технологічному процесі узгоджене і безперервне надходження виробів (деталей) одних за одними на обробку називають потоком, а їхню послідовну обробку – потоковим виробництвом. Засоби виробництва в такому разі розміщують відповідно до технологічної послідовності виконуваних процедур, а комплект цих засобів, розміщених відповідним чином, називають поточною лінією.

Кожний виріб в процесі обробки переміщується між засобами обробки з послідовним чергуванням робочих процедур – транспортних, завантажувальних, виконуваних різними компонентами транспортної системи автоматизованої лінії.

За способами організації транспортних потоків, транспортні системи бувають:

- **синхронні** (жорсткі) транспортні системи – це системи, де всі їхні компоненти працюють у заданій узгодженості з відповідним графіком або тактовим сигналом. Кожна процедура в системі починається і завершується в строго визначений час, а взаємодія між компонентами системи відбувається в точно визначеному порядку. Переваги синхронних транспортних систем полягають у простоті дизайну, легкості управління та передбачуваності роботи;

- **несинхронні** (гнучкі) транспортні системи – це системи, в яких їхні компоненти можуть працювати незалежно один від одного та в будь-якій послідовності. Такі системи не мають суворо визначеного

графіка роботи або тактового сигналу, який контролює час виконання їхніх процедур.

Синхронні автоматичні лінії виробництва поділяють так:

- прямі лінії виробництва – це тип синхронних автоматичних ліній, у яких виробничий процес розділяється на послідовні стадії, а виробник або виробнича лінія переміщується вздовж цих стадій, і кожна частина або складова продукту додається або складається на кожній зі стадій (рис. 8.4);

- лінії переробки – цей тип ліній виробництва використовують для обробки сировини або напівфабрикатів у готові продукти (рис. 8.5). Продукт проходить через різні стадії обробки, такі як очищення, оброблення, пакування тощо, на кожній з яких він піддається різним технологічним процедурам;

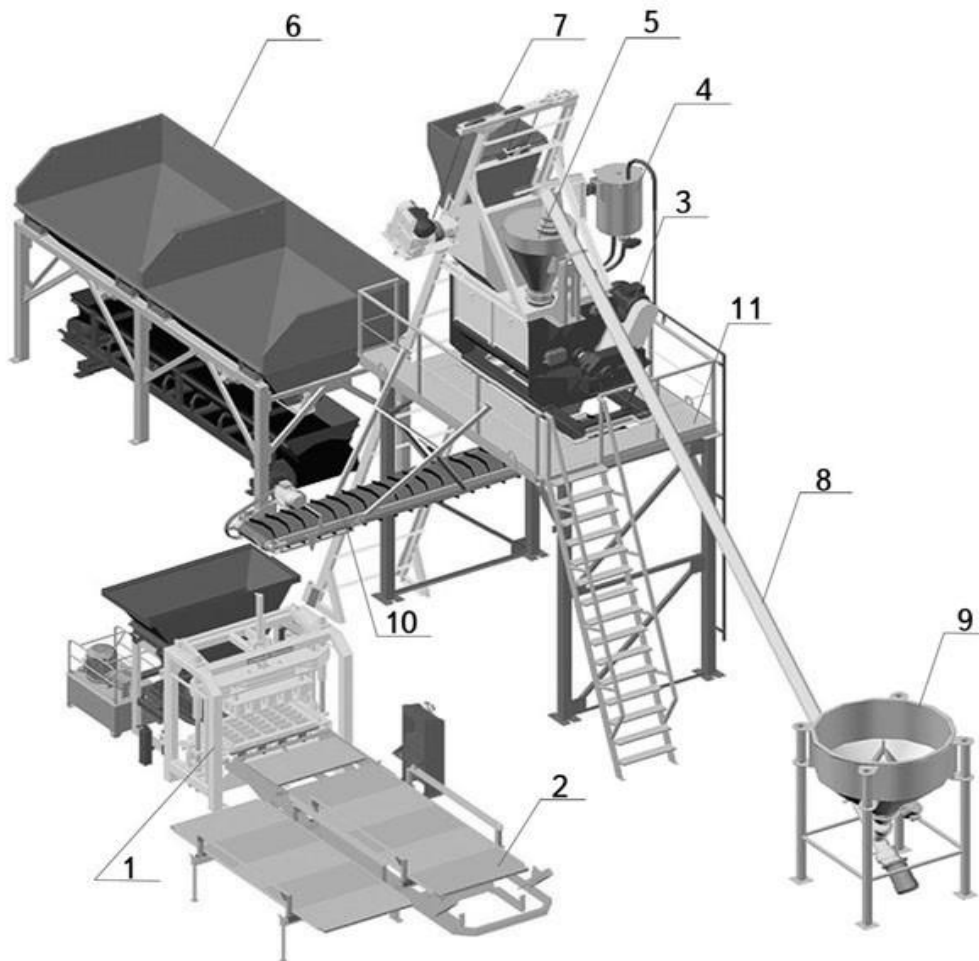


Рис. 8.4. Логістичний процес прямої лінії з виробництва цегли: 1 – формувальний прес; 2 – вібростіл; 3 – привід змішувача; 4 – резервуар з водою; 5 – змішувач; 6 – дозатор; 7 – скіповий підйомник; 8, 9 – живильник сухого цементу; 10 – живильник бетону

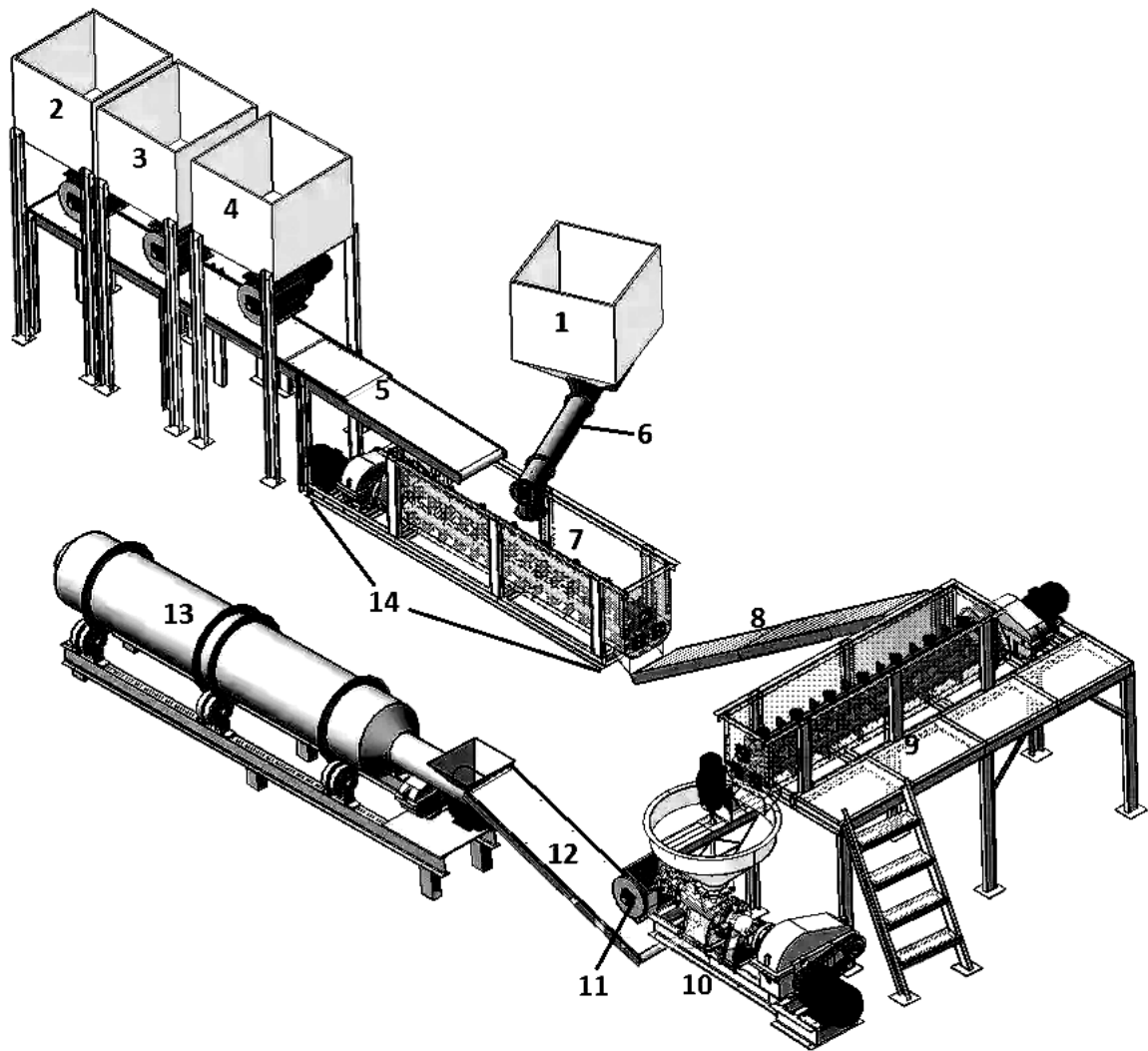


Рис. 8.5. Логістичний процес виробничої лінії переробки вугільних брикетів: 1-4 – бункери з матеріалом; 5, 8, 12 – конвеєр; 6 – живильник; 7, 9 – змішувачі роторні; 10 – формувальний прес; 11 – роздільник; 13 – сушильний барабан; 14 – ваги

– ротаційні лінії – у цьому типі ліній виробництва використовують обертову платформу для переміщення продукту або матеріалу через різні стадії виробництва. Кожна стадія може бути призначена для різних завдань, а продукт обертається на кожній стадії, доки він не досягне кінцевого пункту виробництва (рис. 8.6);

– лінії складання – схожі на пряму лінію виробництва, але складання відбувається за участі працівників або роботів. Робочі місця розміщено вздовж лінії, а кожен працівник або робот виконує певні завдання зі складання продукту, доки він проходить через їхнє робоче місце.

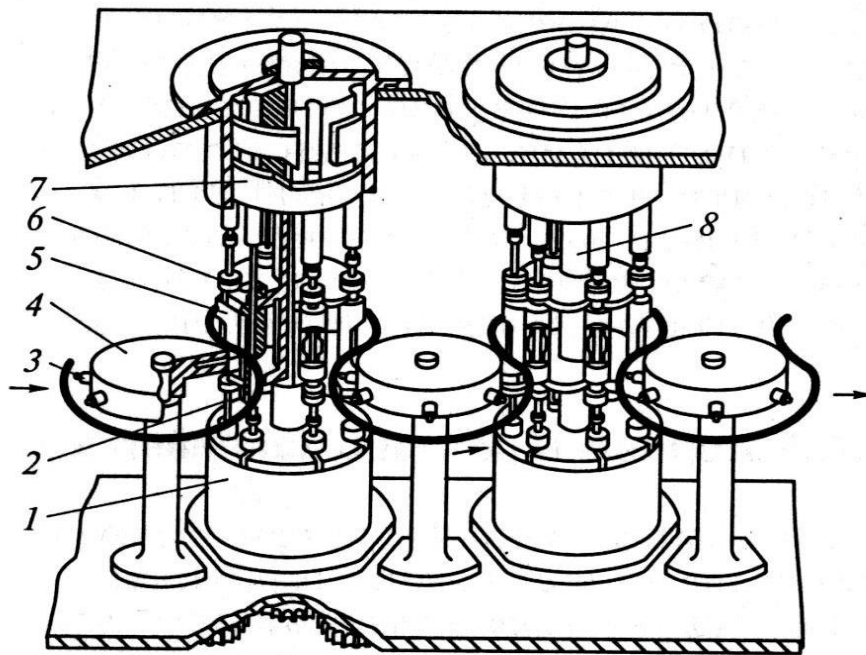


Рис. 8.6. Логістичний процес ротаційної автоматичної виробничої лінії:
1 – технологічний ротор; 2 – штовхач; 3 – кліщі; 4 – транспортний ротор;
5 – супорт; 6 – тримач; 7 – кулачок; 8 – стійка

Розподілення потоків в синхронних транспортних системах відбувається за допомогою ділильників потоку (рис. 8.7) або спеціальних конвеєрів.

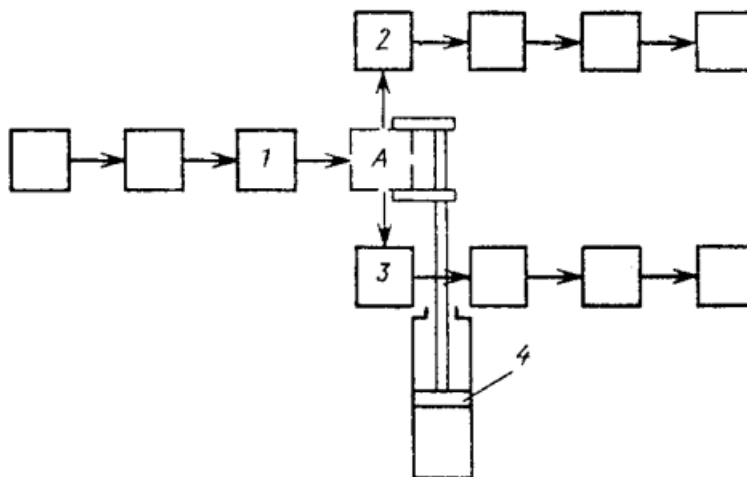


Рис. 8.7. Транспортна система з ділильником потоку

Основні види несинхронних автоматичних ліній виробництва:

- гнучка комірка виробничої системи – ця система складається з набору оброблювальних центрів, роботів та іншого обладнання, які

беруть участь у різних процесах виробництва. Кожна комірка може працювати незалежно, і всі вони можуть бути переналаштовані для виробництва різних продуктів без значних часових витрат;

- модульна виробнича система – у цій системі виробничий процес розділяється на модулі, які можуть бути розміщені та конфігуровані відповідно до потреб виробництва. Кожен модуль здатен працювати незалежно, і їх можна легко додавати або видаляти для зміни виробничих потужностей;
- розподілена виробнича система – це система, у якій використовують мережу розподілених оброблювальних центрів та роботів, які співпрацюють для виконання виробничих завдань. Кожен центр може ухвалювати рішення про обробку чи виробництво самостійно на підставі отриманих вхідних даних;
- адаптивна виробнича система – система, у якій використовують сенсори, штучний інтелект та інші технології для аналізу виробничого процесу та автоматичного налаштування обладнання з метою оптимізації продуктивності та реакції на зміни.

Контрольні запитання

1. Наведіть основні характеристики механічних передач.
2. Що називають передатною функцією?
3. Як визначити крутний момент двигуна мехатронної системи приводу конвеєра?
4. Як визначити крутний момент приводу механічної передачі?
5. Які параметри передачі застосовують для керування її рухом?

Лекція 9. Програмування мехатронних модулів

Програмування мехатронних модулів відбувається із застосуванням їхніх математичних моделей, а виконується контролерами керування. Далі розглянемо пристрої, які здатні здійснювати електронне управління електричними сигналами.

Найпростішим апаратним електронним контролером, який можна застосовувати в мехатронних модулях для програмування та виконання заданих програм є Arduino (рис. 9.1).

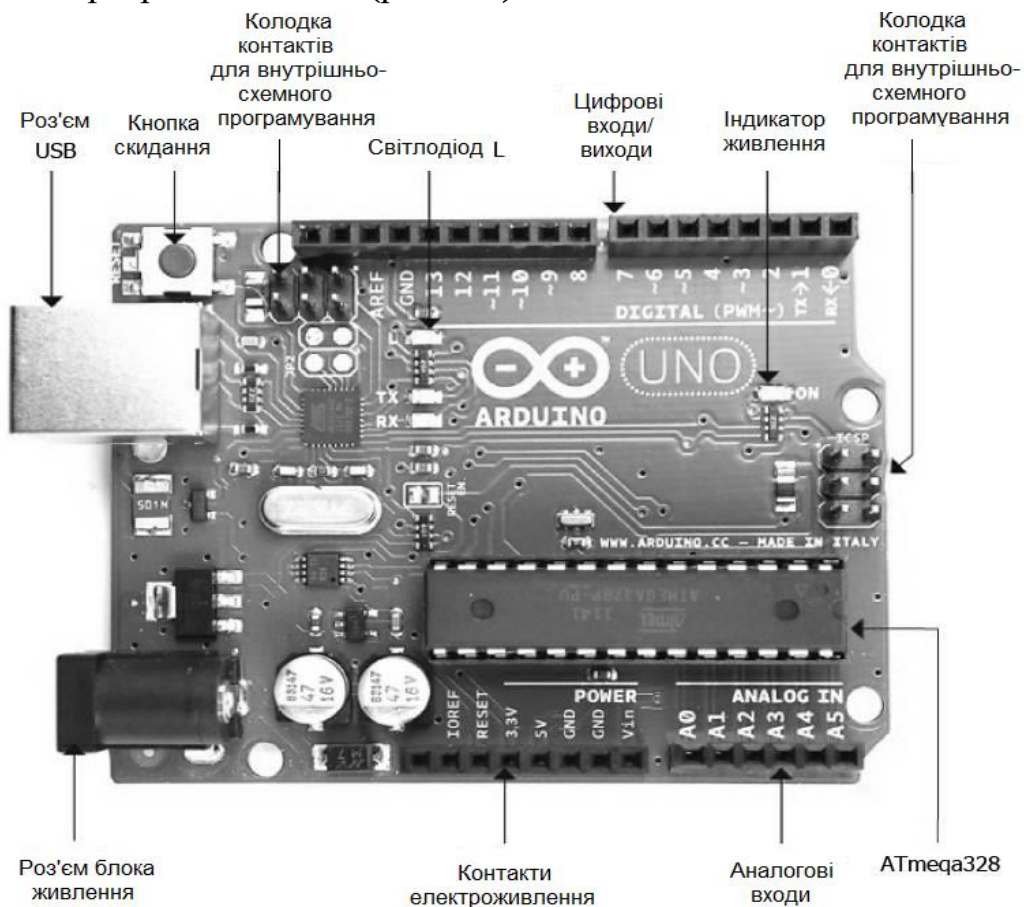


Рис. 9.1. Будова плати Arduino Uno

Плати Arduino сконструйовані для комутації струму на вихідних програмованих контактах, вимірювання значень напруги на вхідних контактах та комунікації з іншими електронними пристроями, які містять відповідні інтерфейси (ICSP, wi-fi). Незалежно від конструктивних особливостей всі плати програмуються мовою C++ в Arduino IDE.

Модель Arduino Uno R3 є останньою в серії стандартних плат, що включає моделі Uno, Duemilanove, Diecimila і NG. Всі ці плати побудовані на основі мікропроцесорів ATmega168 та ATmega328, які

відрізняються лише об'ємом пам'яті. Іншою сучасною моделлю Arduino того ж розміру і з тим самим набором контактів, що й Uno R3, є Arduino Leonardo (рис. 9.2). Плата Leonardo сконструйована на основі процесора ATmega32u4, схожого на ATmega328, але має вбудований інтерфейс USB. Крім того, модель Leonardo містить більше пам'яті і більше аналогових входів.

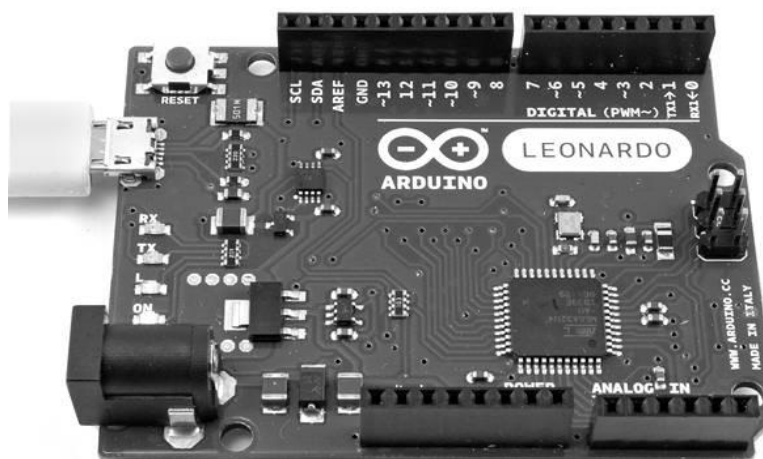


Рис. 9.2. Arduino Leonardo

Якщо кількості контактів вводу/виводу на платах Uno і Leonardo не достатньо для вирішення поставлених завдань, в таких ситуаціях можна застосувати плати розширення, яких доволі багато для Arduino Uno, або застосувати плату більшого розміру. Плати розширення – це додаткові електронні плати, які під'єднують до основної плати і збільшують кількість контактів, але при цьому зменшується продуктивність основної плати. Кількість контактів не може бути збільшена більш ніж це передбачено конструктором та схемою керування.

Найбільшою серед моделей Arduino вважають Arduino Due, яка комплектується процесором із тактовою частотою 84 МГц (порівняно з 16 МГц для моделі Uno), але має проблеми сумісності з іншими моделями. Найбільша з них полягає в тому, що для живлення Due потрібна напруга номіналом 3,3 В замість 5 В, як для більшості моделей Arduino. Arduino Due має ряд переваг, таких, як великий обсяг пам'яті для програм та даних; апаратна підтримка виведення звуків (апаратні цифроаналогові перетворювачі); чотири послідовні порти; два порти USB; інтерфейси USB-хоста та USB OTG; імітація USB-клавіатури та USB-миші.

У платах Arduino використовують мікроконтролери, які виготовляє компанія Atmel і які мають схожу апаратну архітектуру, за винятком

мікроконтролера, використовуваного в моделі Due (SAM3X8E ARM Cortex-M3 CPU).

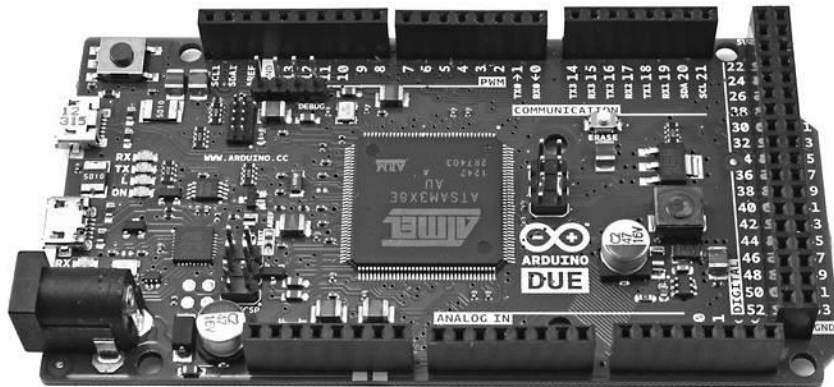


Рис. 9.3. Arduino Due

Серед плат Arduino також є мінімоделі плат. На рис. 9.4 зображено плату Arduino Mini, яка містить інтерфейс USB, а її програмування здійснюється із застосуванням окремого модуля розширення. Окрім Mini є також моделі Nano і Micro. Обидві вони мають інтегрований інтерфейс USB.

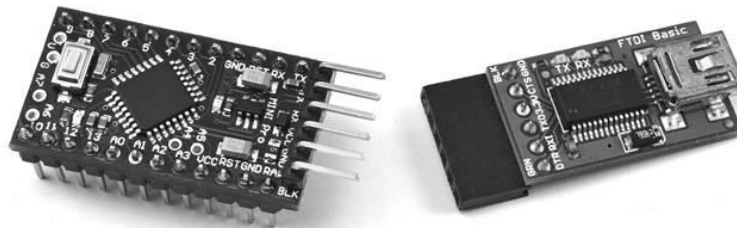


Рис. 9.4. Arduino Mini та Arduino Programmer

Структурну схему плати Arduino Uno зображено на рис. 9.5.

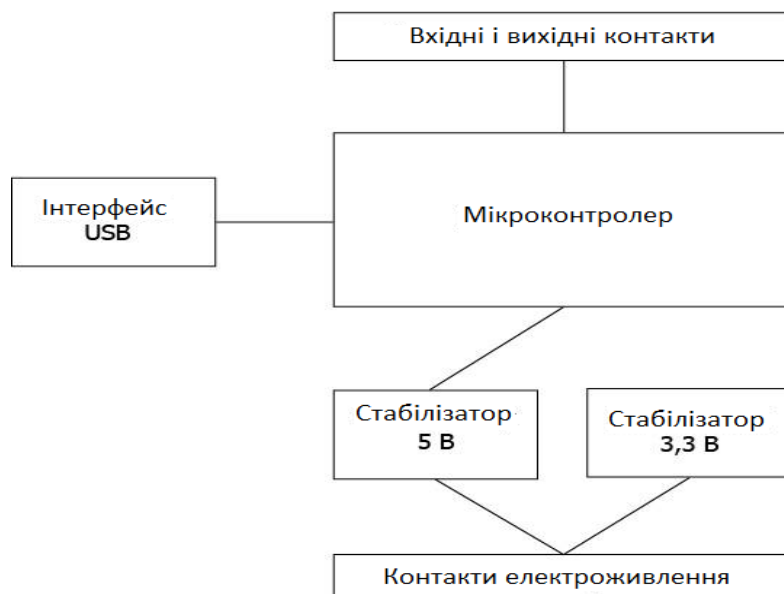


Рис. 9.5. Архітектура плати Arduino Uno

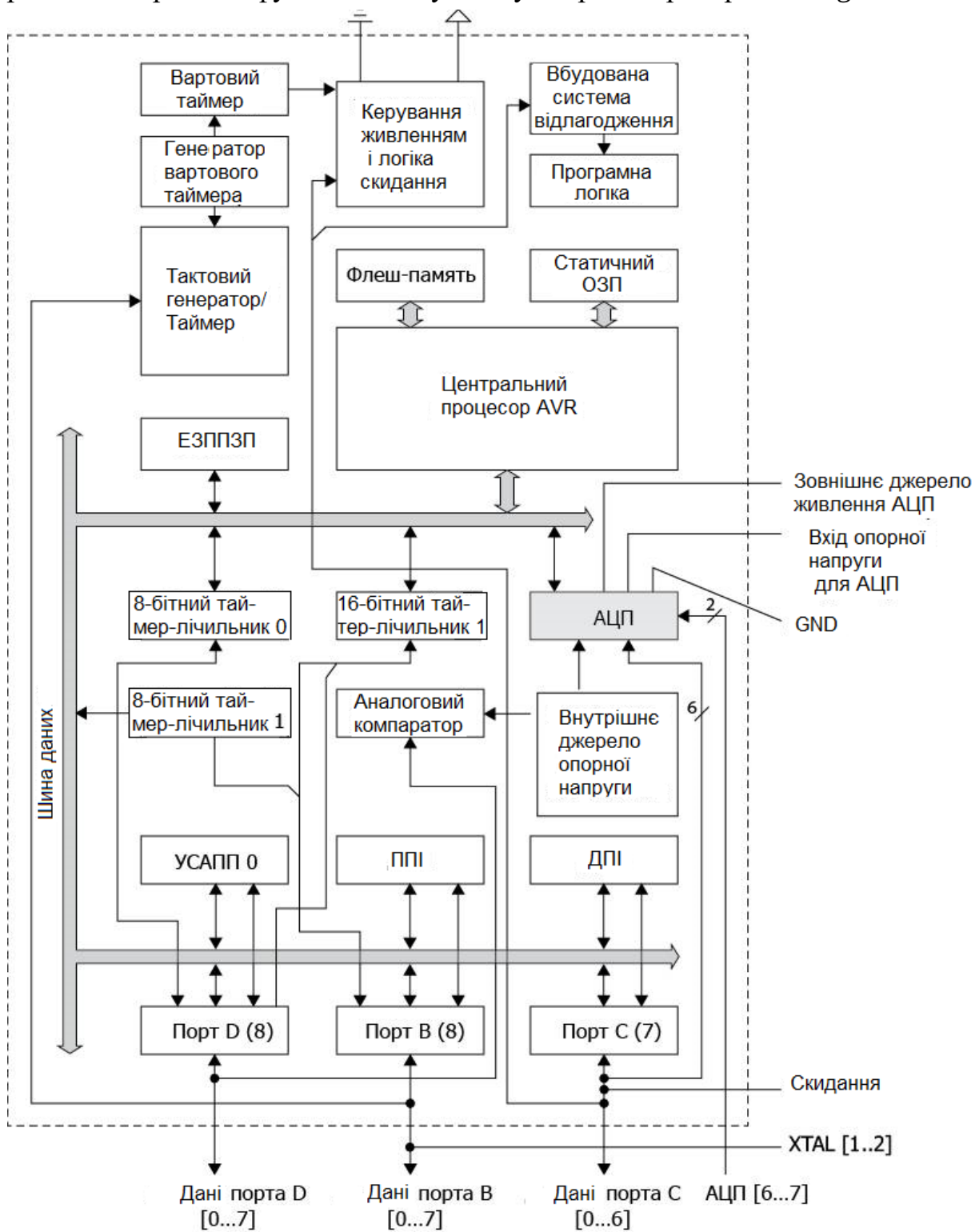


Рис. 9.6. Архітектура мікроконтролера ATmega328

Центральний процесор – це внутрішній пристрій мікроконтролера, який виконує арифметично-логічні процедури. Процесор зчитує всі інструкції (скомпільований код скетчу) із флеш-пам'яті по одній за раз.

Змінні, які є у програмі, зберігаються окремо, в оперативній пам'яті. На відміну від флеш-пам'яті, де зберігається код програми, оперативна пам'ять є енергозалежною та зникає в разі вимкнення живлення. Для даних, які повинні зберігатися навіть після вимикання живлення, слід використовувати пам'ять третього типу – постійну пам'ять, що електрично стирається (Electrically Erasable Programmable Read-Only Memory, EEPROM).

Електроживлення плати Arduino є можливим через рознімач USB або через рознімач зовнішнього блока живлення. На цей рознімач допускається подавати постійну напругу номіналом від 7,5 до 12 В. Плата Arduino споживає близько 50 мА. Після подачі живлення на плату загоряється індикатор живлення.

Контакти на платі призначені для комунікації з пристроями та для живлення плати зокрема. Комунікаційні контакти називають портами вводу/виводу (I/O). Перший контакт без мітки зарезервований для використання в майбутньому. Контакт IOREF призначено для визначення опорної напруги, з якою працює плата. Наприклад, плати Arduino Uno і Leonardo працюють на напрузі 5 В, тому на цьому контакті завжди буде напруга 5 В. Наступний контакт – Reset призначено для скидання програми. При цьому на платі Arduino Uno поруч із рознімачем USB розміщено кнопку скидання, натискання на яку посиляє логічний імпульс на ввід Reset, який у відповідь очистить пам'ять і запустить програму спочатку.

Інші контакти в цій групі слугують для виведення електроживлення з різними рівнями напруги (3.3V, 5V, GND) відповідно до позначень. GND або ground («заземлення») означає 0 В. Контакти GND є опорними точками, відносно яких вимірюється напруга у всіх інших точках на платі. Контакти GND абсолютно ідентичні.

Контакти в наступній групі підписані Analog (аналогові входи) з номерами від 0 до 5. Це шість контактів, які можна використовувати для вимірювання напруги та її аналізу у скетчі. Незважаючи на те, що вони позначені як аналогові входи, їх можна використовувати і як цифрові входи або виходи. За замовчуванням вони діють як аналогові входи.

Контакти, що позначені Digital, є цифровими входами/виходами мікроконтролера. Якщо увімкнути такий контакт зі скетчу, на ньому з'явиться напруга 5 В, а якщо вимкнути – напруга впаде до 0 В. Подібно до контактів електроживлення, ці контакти слід використовувати

обережно, щоб не перевищити максимально допустимий струм. Цифрові виходи можуть віддавати максимальний струм до 40 мА з напругою 5 В.

У мікроконтролерах AVR реалізовано гарвардську архітектуру, відповідно до якої розділено не лише адресні простори пам'яті програм та пам'яті даних, а й шини доступу до них. Кожна з областей пам'яті даних (оперативна пам'ять та EEPROM) також розміщена у своєму адресному просторі.

Пам'ять програм (Flash ROM) призначена для зберігання послідовності команд, що управляють функціонуванням мікроконтролера, вона має 16-бітну організацію. Всі AVR мають Flash-пам'ять програм, яка може бути різного розміру – від 1 до 256 КБайт. Її головна перевага в тому, що вона побудована на принципі електричної перепрограмованості, тобто допускає багаторазове стирання та запис інформації. Після вимкнення живлення ця пам'ять не зникає. Програма контролера заноситься до Flash-пам'яті як з допомогою звичайного програматора, так і за допомогою SPI-інтерфейса, зокрема безпосередньо на зібраній платі. Можливість внутрішньосхемного програмування (функція ISP) через комунікаційний інтерфейс SPI мають усі мікроконтролери AVR, крім Tiny11 і Tiny28. Гарантована кількість циклів перезапису Flash-пам'яті у мікроконтролерів AVR другого покоління становить щонайменше 10 тисяч.

Пам'ять даних розділена на три частини: регістрова пам'ять, оперативна пам'ять (RAM) й енергонезалежна пам'ять (EEPROM).

Регістрова пам'ять має 32 регістри загального призначення (РВН або GPR), об'єднаних у файл, і службові регістри вводу/виводу (РВВ). В області регістрів вводу/виводу розміщені різні службові регістри (реєстри управління мікроконтролером, регістри стану тощо), а також регістри управління периферійними пристроями, що належать до складу мікроконтролера. По суті, управління мікроконтролером полягає в управлінні цими регістрами.

Внутрішня оперативна статична пам'ять Static RAM (SRAM) має байтовий формат, призначена для оперативного зберігання даних. Розмір оперативної пам'яті може змінюватись у різних чипів від 64 Байт до 4 КБайт. Кількість циклів читання і запису в RAM не обмежена. Для деяких мікроконтролерів можлива організація під'єднання зовнішнього статичного ОЗУ обсягом до 64К.

У моделях Arduino Leonardo та LilyPad USB, а також Micro та Nano використовується мікроконтролер ATmega32u4B, що схожий на ATmega328, але має більш сучасний чип з кількома доповненнями, яких немає в ATmega328, а саме:

- вбудований інтерфейс USB, завдяки чому відпала потреба у додаткових апаратних компонентах підтримки USB;
- збільшено кількість контактів ШІМ;
- наявність двох послідовних портів;
- окремі контакти інтерфейсу I2C;
- об'єм статичної оперативної пам'яті, більший на 0,5 Кбайт.

Мікроконтролер ATmega2560, використовуваний в моделях Arduino Mega 2560 та Arduino Mega ADK, має більше пам'яті кожного типу (256 Кбайт флеш-пам'яті, 8 Кбайт статичного RAM та 4 Кбайт EEPROM), а також набагато більше контактів введення/виводу.

На рис. 9.7 наведено деякі пристрої, які можна приєднувати до плати Arduino.

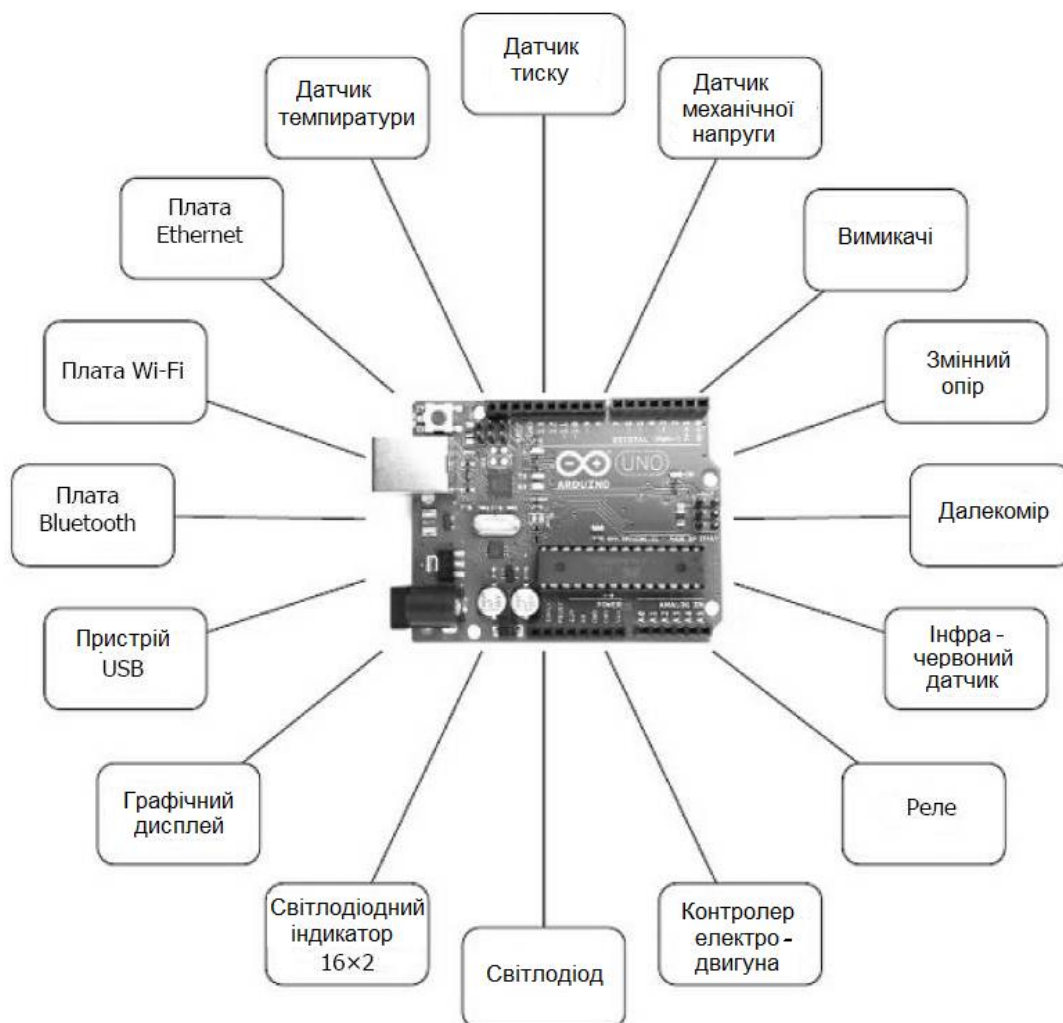


Рис. 9.7. Пристрої, які можна під'єднувати до Arduino

Для програмування мікроконтролерів застосовують мови програмування, які можуть працювати з внутрішньою фізичною пам'яттю мікроконтролерів на низькому абстрактному рівні (на рівні машинного коду). Можуть застосовуватися й інші мови програмування, але з використанням спеціальних бібліотек, які надаватимуть доступ до пам'яті мікроконтролера. Мікроконтролери Atmega328 на платах Arduino програмують мовами Cі та C plus, проте для простоти програмування компанія Arduino пропонує застосовувати їхню власну мову Wiring та середовище розробки Arduino IDE. Оскільки мікроконтролери мають обмежений об'єм пам'яті, практично не застосовують об'єктно орієнтовані підходи, а більшість програм пишуть у функціональному процедурному стилі.

Для встановлення середовища розробки Arduino IDE слід завантажити та встановити програмне забезпечення для своєї операційної системи з офіційного веб-сайту Arduino. Щоби переконатися в працездатності плати, запишемо в неї програму, яка вмикатиме і вимикатиме світлодіод, позначений на платі Arduino міткою L. Запустіть Arduino IDE на комп'ютері, потім у меню **File (Файл)** (рис. 9.8) оберіть пункт **Examples → 01 Basics → Blink**.

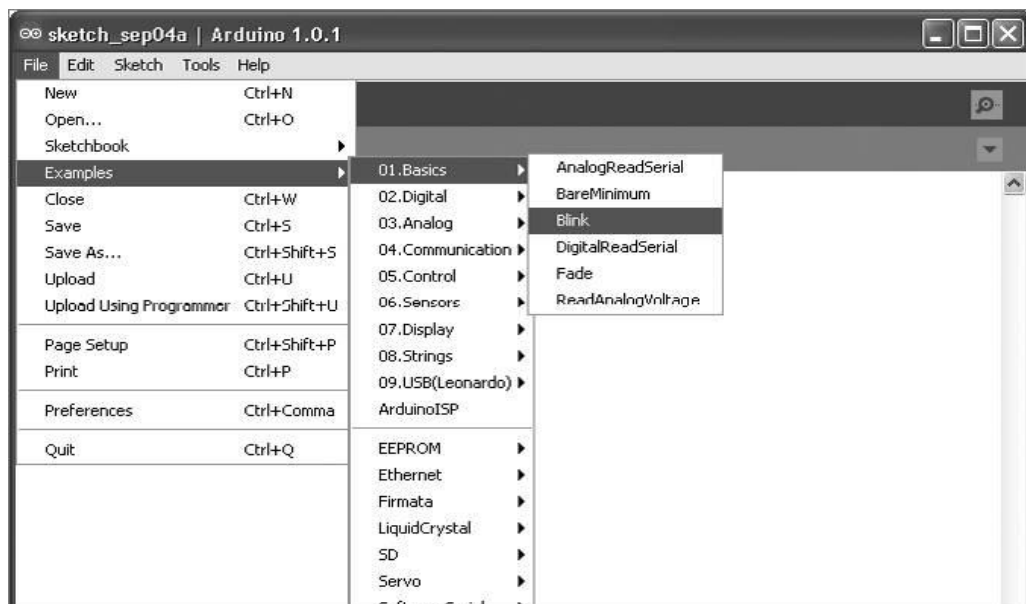


Рис. 9.8. Завантаження скетча Blink в Arduino IDE

Програми Arduino називають скетчами. Перш ніж вивантажити скетч **Blink** у вашу плату Arduino, потрібно налаштувати в Arduino IDE тип плати. Розглянемо приклад налаштування для плат Arduino Uno, тому в меню **Tools → Board** виберіть **Arduino Uno** (рис. 9.9).

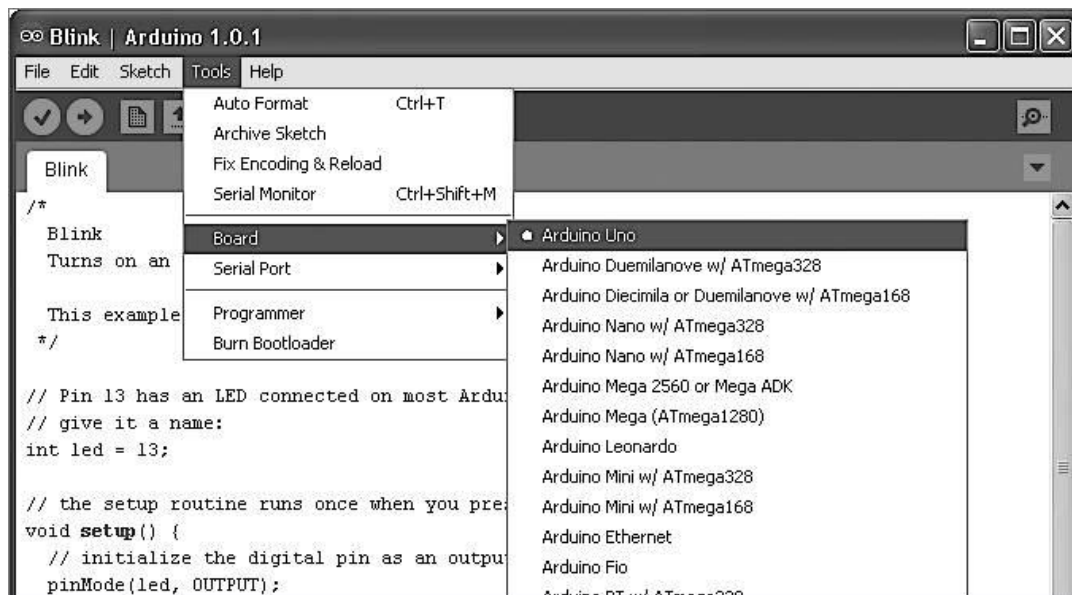


Рис. 9.9. Вибір типу плати

Після налаштування типу плати потрібно обрати порт під'єднання плати. Середовище розробки Arduino IDE поміщає останні під'єднані пристрої на початку списку, тому плата Arduino має бути вгорі (рис. 9.10).

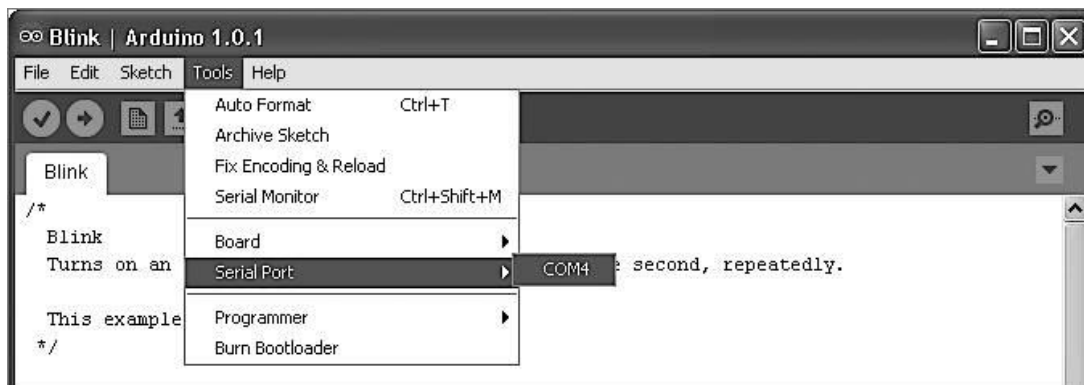


Рис. 9.10. Вибір послідовного порту

Для завантаження скетчу на плату Arduino, потрібно натиснути Upload (Завантаження) на панелі інструментів – другий ліворуч (підсвічено на рис. 9.11). Після цього в Arduino IDE з'явиться індикатор, який відображатиме перебіг компіляції скетчу (тобто його перетворення у формат для контролера), далі деякий час повинні блимати світлодіоди з мітками Rx та Tx на платі Arduino, і після цього повинен почати блимати світлодіод з міткою L. В нижній частині вікна Arduino IDE також з'явиться інформація про завантаження скетчу в Arduino.



Рис. 9.11. Завантаження скетчу Blink

Blink – це найпростіша програма, написана мовою Wiring (це доповнена спеціальними бібліотеками Arduino мова програмування C), яка виконує ввімкнення світлодіода на одну секунду, потім вимикає його на одну секунду, і це повторюється в нескінченному циклі. Перша частина скетчу – це коментар, що описує призначення скетчу. Коментар не є програмним кодом. У процесі підготовки коду (компіляції) до завантаження в мікроконтролер всі коментарі видаляються. Коментарі можуть бути однорядковими та багаторядковими. Все, що є між парами символів «/*» та «*/», вважається багаторядковим коментарем та ігнорується комп'ютером. Однорядкові коментарі починаються із символу «//». Ці коментарі описують те, що відбувається.

В програмі Blink під'єднано контакт № 13, тому що на платі Arduino Uno саме цей контакт стандартно паралельно під'єднано до світлодіода L:

```
// На більшості плат Arduino до контакту 13 підключено світлодіод.
// Дамо йому ім'я:
int led = 13;
```

Наступна частина скетчу – функція **setup**. Ця функція повинна бути в кожному скетчі, і вона виконується щоразу, коли відбувається

перезавантаження плати Arduino, або в результаті натискання на кнопку скидання Reset, або після подачі електроживлення на плату:

```
// процедура setup виконується один раз після натискання кнопки скидання
void setup() {
  // ініціалізувати контакт як цифровий вихід
  pinMode(led, OUTPUT);
}
```

Функція – це фрагмент програмного коду, який має власне ім'я (у цьому випадку **setup**), далі повинен бути символ «{», за яким ідуть потрібні команди, кожна в окремому рядку, що завершуються крапкою з комою (;). Кінець функції позначається символом «}».

У наведеному прикладі Arduino виконає єдину команду **pinMode(led, OUTPUT)**, яка налаштовує контакт працювати у режимі виходу.

Далі за кодом зазначають основну частину скетчу – функцію **loop**. За аналогією з функцією **setup**, кожен скетч повинен містити функцію **loop**. Але на відміну від функції **setup**, яка виконується лише один раз після скидання, **loop** виконується знову і знову. Тобто тільки-но будуть виконані всі її інструкції, вона відразу запускається заново.

Функція **loop** вмикає світлодіод, виконуючи інструкцію **digitalWrite(led, HIGH)**, потім виконується команда **delay(1000)**, яка зупиняє скетч на 1 с (значення 1 000 означає тут 1 000 мс, або 1 с). Після цього виконується інструкція **digitalWrite(led, LOW)**, і світлодіод вимикається, скетч зупиняється ще на 1 с командою **delay(1000)**, і процес повторюється:

```
void loop() {
  // ввімкнути контакт led встановивши високий рівень напруги (HIGH)
  digitalWrite(led, HIGH);
  delay(1000); // Чекати 1 с
  // вимкнути контакт led встановивши низький рівень напруги (LOW)
  digitalWrite(led, LOW);
  delay(1000); // Чекати 1 с
}
```

Щоб збільшити частоту миготіння світлодіода, потрібно змінити обидва числа «1000» на число «200» в інструкції **delay**.

Якщо спробувати зберегти скетч перед розвантаженням, Arduino IDE нагадає, що цей скетч є прикладом і доступний тільки для читання, але запропонує зберегти копію в стандартному каталозі для скетчів (**File → Sketchbook**) Arduino IDE.

Якщо натиснути Upload в Arduino IDE, скетч буде завантажений в контролер, плата Arduino автоматично перезапуститься і почне працювати згідно з програмою завантаженого скетча.

Змінні допомагають зробити програму управління конфігурованою, а також застосовуються для роботи з даними. Оголошуючи змінну у мові Cі, треба вказати її тип. Наприклад, якщо потрібна змінна, що містить ціле число, її слід оголосити з типом `int` (ціле зі знаком), тоді щоб визначити змінну з іменем `delayPeriod` та значенням 200, потрібно записати таке:

```
int delayPeriod = 200;
```

`delayPeriod` – це ім'я, тому в ньому не може бути пробілів. За загальними угодами імена змінних повинні починатися з літери в нижньому регістрі, а кожне нове слово в імені – з літери у верхньому регістрі.

Зазвичай рядки програми виконуються послідовно, один за одним, без винятків, але якщо потрібно змінити послідовність виконання лише за певної умови, застосовують процедури розгалуження, наприклад:

```
void setup() {  
    pinMode(5, INPUT_PULLUP);  
    pinMode(9, OUTPUT);  
}  
void loop() {  
    if (digitalRead(5) == LOW) {  
        digitalWrite(9, HIGH);  
    }  
}
```

У цьому прикладі умова (після оператора **if**) виконується, якщо з контакту 5 прочитано значення LOW. Два знаки «`==`» позначають процедуру порівняння двох значень (не плутати з єдиним знаком «`=`», що позначає процедуру надання значення змінної). Оператор **if** каже: якщо умова є істинною (тобто виконується), то повинні бути виконані команди у фігурних дужках (в наведеному прикладі команда, що встановлює рівень напруги на цифровому виході HIGH 9). Якщо умови не дотримано, контролер не виконуватиме блок в **If** і далі виконуватиме програму. У такому разі буде досягнутий кінець функції `loop`, і програма запуситься наново.

За аналогією з виконанням деяких процедур за умовою іноді в скетчі виникає потреба повторювати процедури. Звичайно така поведінка вже доступна у функції **loop**, але іноді потрібно виконати команди кілька разів всередині **loop**. Домогтися цього можна за допомогою команди `for`, що дає змогу використовувати змінну-лічильник. Наприклад, напишемо скетч, який блимає світлодіодом 10 разів:

```
// sketch 01_01_blink_10  
int ledPin = 13;  
int delayPeriod = 200;
```

```

void setup() {
  pinMode(ledPin, OUTPUT);
}
void loop() {
  for (int i = 0; i < 10; i++) {
    digitalWrite(ledPin, HIGH);
    delay(delayPeriod);
    digitalWrite(ledPin, LOW);
    delay(delayPeriod);
  }
}

```

Команда «**for**» визначає змінну з ім'ям «**i**» та надає їй початкове значення 0, далі після символу «**;**» зазначається умова «**i<10**», яка означає умову продовження циклу (доки значення «**i**» залишається меншим, ніж 10, виконуються команди, укладені у фігурні дужки). Останній елемент **for** – «**i++**», що означає «**i=i+1**» (збільшення лічильника «**i**» на 1). Збільшення значення «**i**» на одиницю відбувається наприкінці кожного циклу.

Функції – це спосіб групування програмних команд у зручний для використання блок. Функції допомагають поділити скетч на керовані та прості у використанні фрагменти. Наприклад, напишемо скетч, який спочатку швидко блимає світлодіодом 10 разів, а потім починає блимати із частотою один раз на секунду:

```

int ledPin = 13;
void setup() {
  pinMode(ledPin, OUTPUT);
  flash(10, 100);
}
void loop() {
  flash(1, 500);
}
void flash(int n, int delayPeriod) {
  for (int i = 0; i < n; i++) {
    digitalWrite(ledPin, HIGH);
    delay(delayPeriod);
    digitalWrite(ledPin, LOW);
    delay(delayPeriod); } }

```

Визначення функції знаходиться наприкінці скетчу. Перший рядок у визначенні функції **flash (int n, int delayPeriod)** вказує Arduino, що визначається нова функція з ім'ям **flash**, яка приймає два параметри, обидва типи **int**. Перший параметр, з ім'ям **n**, визначає, скільки разів світлодіод повинен мигнути, а другий, з ім'ям **delayPeriod**, визначає величину паузи після ввімкнення або вимкнення світлодіода. Ці два параметри можна використовувати лише всередині функції, тому **n** використовується в команді **for**, де визначає кількість повторень циклу, а **delayPeriod** – всередині команд **delay**. Функція **loop** скетчу запускає

функцію **flash** із заданими параметрами. Оскільки ця команда міститься всередині функції **loop**, вона буде виконуватися знову і знову, змушуючи світлодіод блимати безперервно.

Розглянемо програму керування Arduino цифровим контактом 7 (рис. 9.12):

```
int ledPin = 13;
int switchPin = 7;
void setup() {
  pinMode(ledPin, OUTPUT);
  pinMode(switchPin, INPUT_PULLUP);
}
void loop() {
  if (digitalRead(switchPin) == LOW) {
    flash(100);
  }
  else {
    flash(500);
  }
}
void flash(int delayPeriod) {
  digitalWrite(ledPin, HIGH);
  delay(delayPeriod);
  digitalWrite(ledPin, LOW);
  delay(delayPeriod);
}
```

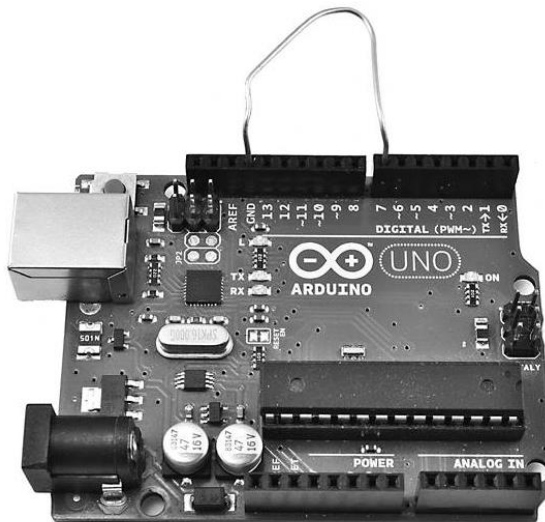


Рис. 9.12. Дослід роботи з цифровим контактом 7

Використовуючи провідник, замкніть контакти **GND** і 7, як показано на рисунку. Це можна робити тільки після вивантаження розглянутого вище скетчу, що пов'язане з тим, що попередній скетч міг налаштувати контакт 7 на роботу в режимі виходу – у такому разі замикання на **GND** може пошкодити контакт. Оскільки скетч налаштовує контакт 7 на роботу в режимі входу, його безпечно з'єднувати з контактом **GND**. Згідно з цим скетчем відбуватиметься таке: коли контакти замкнуті, світлодіод блимає частіше, а коли не замкнуті – рідше.

У скетчі є змінна з ім'ям **switchPin**, яка символізує номер контакту 7. Скрипка у цьому експерименті відіграє роль вимикача (можна застосувати кнопку). У функції **setup** контакт 7 налаштовується працювати у режимі входу командою **pinMode**. Другим аргументом процедури **pinMode** передається значення **INPUT_PULLUP**, яке означає, що за замовчуванням вхід повинен мати рівень напруги **HIGH**, та якщо на нього не подано рівень **LOW**, з'єднуємо цей контакт з контактом **GND**. У функції **loop** використовується команда **digitalRead** для перевірки рівня напруги на вхідному контакті. Якщо цей рівень дорівнює **LOW** (провідник замикає контакти), запускається функція з ім'ям **flash** та значенням 100 (у параметрі **delayPeriod**), що призведе до частого блимання світлодіода. Якщо вхідний контакт має рівень напруги **HIGH**, виконуються команди розділу **else** інструкції **if**, де запускається також функція **flash**, але з більш тривалою паузою, що змушує світлодіод блимати рідше.

Цифрові входи можуть з'єднуватися з цифровими виходами інших модулів, які діють не так, як вимикач, але встановлюють однакові напруги **HIGH** і **LOW**. У таких випадках функції **pinMode** слід передавати аргумент **INPUT** замість **INPUT_PULLUP**.

Налаштування цифрових контактів на роботу в режимі цифрових виходів відбувається у функції **setup** за допомогою команди **pinMode (outputPin, OUTPUT)**. Для того щоби на цифровому виході встановити рівень напруги **HIGH** або **LOW**, потрібно запустити команду **digitalWrite**.

Оскільки плата Arduino під'єднується до комп'ютера через порт USB, можна пересилати дані між комп'ютером і контролером, використовуючи компонент Arduino IDE, який називається монітором послідовного порту (**Serial Monitor**). Для ілюстрації розглянемо скетч, в якому замість зміни частоти миготіння світлодіода після встановлення рівня напруги **LOW** на цифровому вході 7 відбувалося надсилання даних на комп'ютер:

```
int switchPin = 7;
void setup() {
  pinMode(switchPin, INPUT_PULLUP);
  Serial.begin(9600);
}
void loop() {
  if (digitalRead(switchPin) == LOW) {
    Serial.println("Paperclip connected");
  }
  else {
    Serial.println("Paperclip NOT connected");
  }
}
```

```

}
delay(1000);
}

```

Не від'єднуючи плати Arduino від USB і комп'ютера, слід відкрити монітор послідовного порту в Arduino IDE (рис. 9.13). Після роз'єднання контактів на Arduino, в моніторі порту повинен змінитися текст відображення. Для роботи з послідовним портом в Arduino застосовують команду **Serial.begin**, яка ініціалізує передачу даних мікроконтролером. Щоби записати дані в послідовний порт, достатньо виконати команду **Serial.println**.



Рис. 9.13. Монітор послідовного порту

Масиви в програмуванні призначені для збереження списків значень, на відміну від змінних, які можуть зберігати лише одне значення. Масив надає окремі значення зі списку через їхні позиції в такому списку.

У мові Сі, як і більшості інших мов програмування, нумерація позицій у масиві починається з 0, а не з 1. Це означає, що перший елемент фактично є нульовим елементом. В масивах у мові Сі можуть бути дані одного типу, наприклад, числа, символи, булеві значення.

Наступний скетч має масив символів і обирає їх по одному у випадковому порядку та надсилає в монітор послідовного порту через випадкові інтервали часу:

```

char* messages[ ] = {
    "My name is Arduino",
    "Buy books by Simon Monk",
    "Make something cool with me",
    "Raspberry Pis are fruity" };
void setup() {
    Serial.begin(9600);
}
void loop() {
    int delayPeriod = random(2000, 8000);
    delay(delayPeriod);
    int messageIndex = random(4);
    Serial.println(messages[messageIndex]);
}

```

Рядки часто називають колекцією символів, який має тип **char***. Символ зірочки (*) означає, що це вказівник на щось. Квадратні дужки «[]» наприкінці оголошення змінної вказують на те, що ця змінна зберігає масив даних типу **char*** або єдине значення **char***.

Всередині функції **loop** змінної **delayPeriod** надається випадкове значення з діапазону від 2 000 до 7 999 (другий аргумент **random** не входить в діапазон). Потім запуском функції **delay** виконується пауза, тривалість якої дорівнює отриманому проміжку. Змінній **messageIndex** також надається випадкове значення за допомогою команди **random**, але на цей раз їй передається єдиний параметр, внаслідок чого вона повертає випадкове число в діапазоні від 0 до 3, яке потім використовується як індекс повідомлення для відправки до монітора порту.

Контакти з позначками від **A0** до **A5** на платі Arduino можна використовувати для вимірювання напруги, що на них подається. Рівень напруги повинен бути в діапазоні від 0 до 5 В. Вимірювання виконується за допомогою вбудованої функції **analogRead**, яка повертає значення в діапазоні від 0 до 1023: значення 0 є відповідним напрузі 0 В, а значення 1023 – напрузі 5 В. Тобто, щоб перетворити число значення в діапазоні від 0 до 5, потрібно розділити отримане число на 5: $1023/5 = 204,6$. Тип даних **int** не дуже добре підходить для вимірювання напруги, тому що представляє лише цілі числа, а бажано було б також фіксувати дробову частину. Для цього слід використовувати тип даних **float**.

Проаналізуємо наступний скетч Arduino із замкненими контактами A0 та 3.3V (рис. 9.14):

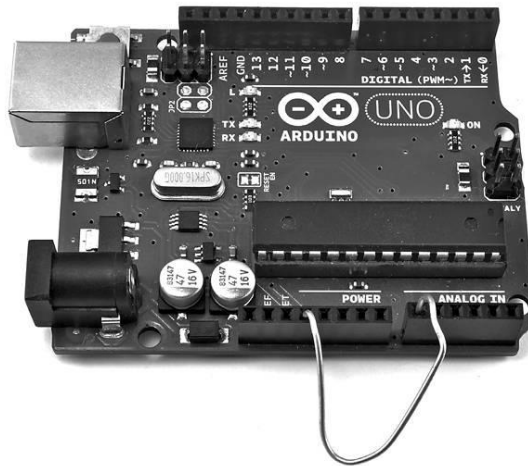


Рис. 9.14. З'єднання контактів A0 і 3.3V

```
int analogPin = A0;
void setup() {
  Serial.begin(9600);
}
void loop() {
  int rawReading = analogRead(analogPin);
  float volts = rawReading / 204.6;
  Serial.println(volts);
  delay(1000);
}
```

Відкривши монітор послідовного порту, можна побачити потік чисел, близьких до значення 3,3 (рис. 9.15) (заборонено замикати контакти електроживлення 5V, 3.3V та GND).

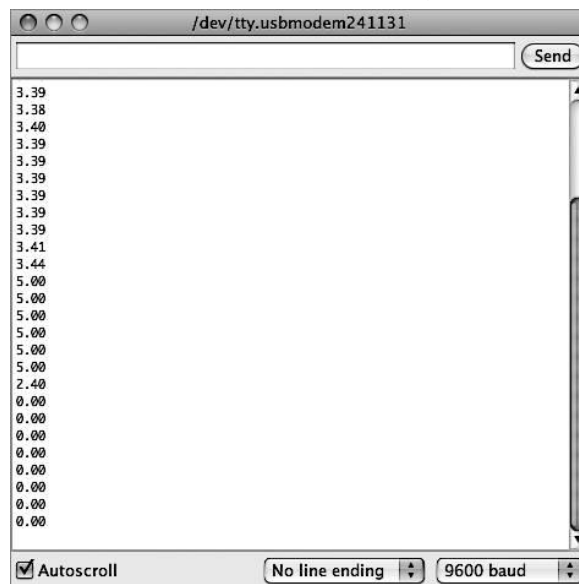


Рис. 9.15. Виведення значень напруги

Плата Arduino Uno не має справжніх аналогових виходів, але має декілька виходів з широтно-імпульсною модуляцією (Pulse-Width Modulation, PWM), які можуть імітувати аналогові виходи, керуючи тривалістю імпульсів (рис. 9.16).

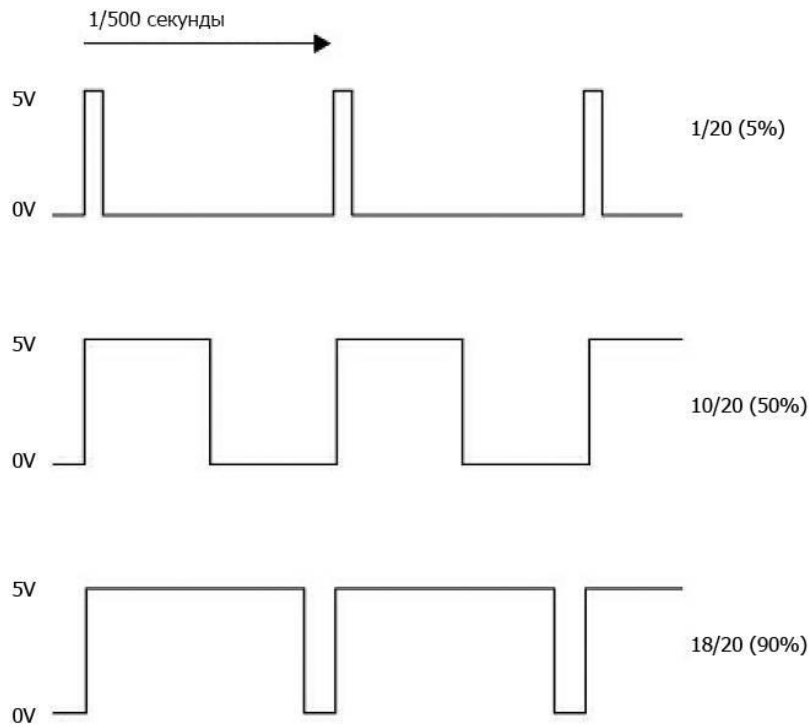


Рис. 9.16. Широтно-імпульсна модуляція

Що довший позитивний імпульс, то вища середня напруга на виході. Оскільки імпульси йдуть з частотою 500 разів за секунду, а більшості пристроїв не властива миттєва реакція, виникає ефект зміни напруги. Контакти, позначені на платі Arduino Uno значком ~ (контакти 3, 5, 6, 9, 10 та 11), є PWM-контактами, їх можна використовувати як аналогові виходи.

Проаналізуємо наступний скетч для PWM сигналу під час роботи з контактом 6 Arduino Uno.

```
int pwmPin = 6;
void setup()
{
  pinMode(pwmPin, OUTPUT);
  Serial.begin(9600);
}
void loop()
{
  if (Serial.available())
  {
    int dutyCycle = Serial.parseInt();
    analogWrite(pwmPin, dutyCycle);
  }
}
```

Відкрийте монітор послідовного порту та введіть число в діапазоні від 0 до 255 у текстове поле у верхній частині вікна, ліворуч від кнопки **Send** (Надіслати). Після натискання кнопки **Send** на під'єднаному до контакту 6 мультиметрі має зміниться напруга. Якщо надіслати число 0,

напруга повинна дорівнювати 0. Відправлення числа 127 має дати середину між 0 і 5 (2,5 В), а число 255 має дати напругу близько 5 В. У цьому скетчі функція **loop** починається з оператора **if**. Умовою для **if** є результат виконання команди **Serial.available()**, тобто якщо монітор послідовного порту отримав значення, виконуються команди всередині фігурних дужок. У цьому випадку команда **Serial.parseInt** перетворює текст рядка, введеного у вікні монітора порту в значення типу **int**, яке потім передається як аргумент у функцію **analogWrite** для виведення імпульсів на контакт.

Через те що плати Arduino мають дуже обмежений об'єм пам'яті, є сенс включати в програму, яка в кінцевому підсумку опиниться на платі, тільки той код, який дійсно буде потрібний. Один із способів досягти цього – використовувати бібліотеки. В Arduino під бібліотекою розуміють колекцію функцій. Наприклад, Arduino IDE містить бібліотеку для роботи зі світлодіодним рідкокристалічним дисплеєм. Вона займає приблизно 1,5 Кбайт пам'яті програм. Немає сенсу під'єднувати цю бібліотеку, якщо її не використовують, тому такі бібліотеки під'єднують лише за потреби.

Під'єднання виконують, додаючи директиви **#include** на початку скетчу. Додати інструкції **include** для під'єднання будь-яких бібліотек, що поставляються в Arduino IDE, можна за допомогою пунктів меню **Sketch → Import Library**.

До складу Arduino IDE входить велика колекція офіційних бібліотек, зокрема:

- EEPROM – для збереження даних в програмно-постійну пам'ять, що електрично стирається (ЕСППЗУ) (Electrically Erasable Programmable Read-Only Memory);
- Ethernet – для реалізації мережних взаємодій;
- Firmata – стандартна бібліотека для взаємодій через послідовний порт;
- LiquidCrystal – для роботи з алфавітно-цифровими рідкокристалічними дисплеями;
- SD – для читання та записування даних на картах флеш-пам'яті;
- Servo – для керування сервоприводами;
- SPI – для реалізації взаємодій по шині послідовного периферійного інтерфейсу;

- Software Serial – для реалізації взаємодій щодо послідовних ліній з використанням будь-яких цифрових виходів;
- Stepper – для керування кроковими електромоторами;
- WiFi – щоб отримати доступ до бездротової мережі WiFi;
- Wire – для реалізації взаємодій з периферією за протоколом I2C.

Деякі бібліотеки призначені для конкретних моделей плат Arduino:

- Keyboard – дозволяє платам Arduino імітувати USB-клавіатуру (Leonardo, Due та Micro);
- Mouse – дозволяє платам Arduino імітувати USB-мишу (Leonardo, Due та Micro);
- Audio – утиліти для відтворення звуку (тільки Due);
- Scheduler – для керування виконанням кількох потоків (тільки Due);
- USBHost – для під'єднання периферії USB (тільки Due).

Нарешті, є безліч бібліотек, написаних іншими користувачами Arduino, які можна завантажити з Інтернету. Далі названо деякі особливо популярні:

- OneWire – для читання даних із цифрових пристроїв з інтерфейсом 1-wire, які випускає компанія Dallas Semiconductor;
- Xbee – для реалізації бездротових взаємодій;
- GFX – графічна бібліотека для роботи з різними дисплеями, які випускає компанія Adafruit;
- Capacitive Sensing – для роботи з ємнісними датчиками;
- FFT – бібліотека частотного аналізу.

Якщо потрібно використовувати одну із сторонніх бібліотек, її потрібно встановити, завантаживши та зберігши в теці Libraries, що міститься в теці Arduino (Documents (Документи)). Зверніть увагу: якщо теки Libraries немає, її спочатку потрібно створити і лише потім додавати до неї бібліотеки.

Для змінних типу **int** в Arduino C відведено 2 байти пам'яті. Якщо тільки скетч не ставить особливих вимог до ощадливого використання пам'яті, значення **int** придатне практично для будь-яких видів інформації, навіть для логічних значень і невеликих цілих чисел, які можна зберігати в однобайтовому значенні.

Повний перелік доступних типів даних наведено у табл. 9.1.

Таблиця 9.1

Типи даних Arduino C

Тип	Займана пам'ять, байт	Діапазон значень	Примітки
boolean	1	true або false (1 або 0)	Використовується для представлення логічних значень
char	1	-128 ... +128	Використовується для представлення кодів символів ASCII, наприклад, А має код 65. Негативні значення зазвичай не використовують
byte	1	0...255	Часто використовується як елементарна одиниця даних під час обміну через послідовні інтерфейси.
int	2	-32 768 ... +32 767	Цілі 16-бітові значення зі знаком
unsigned int	2	0...65 535	Використовується збільшення точності в розрахунках, де використовують негативні числа. Застосовуйте з обережністю, тому що під час використання у виразах спільно зі значеннями типу int можуть виходити несподівані результати
long	4	-2 147 483 648 ... + 2 147 483 647	Потрібен лише для представлення дуже великих чисел
unsigned long	4	0...4 294 967 295	Див. опис типу unsigned int
float	4	-3,4028235E +38 ... +3,4028235E +38	Використовується для представлення дійсних чисел
double	4		в Arduino тип double є повним аналогом типу float

У бібліотеці Arduino є велика кількість команд. В табл. 9.2 зазначено найчастіше використовувані функції Arduino.

Таблиця 9.2

Функції з бібліотеки Arduino

Команда	Приклад	Опис
Цифрове введення/виведення		
pinMode	pinMode(8, OUTPUT);	Переводить контакт 8 у режим цифрового виходу. Підтримуються також режими INPUT та INPUT_PULLUP
digitalWrite	digitalWrite(8, HIGH);	Встановлює високий рівень напруги на контакті 8. Щоб встановити низький рівень напруги, використовуйте константу LOW замість HIGH
digitalRead	int i; i = digitalRead(8);	Надає змінній значення HIGH чи LOW залежно від рівня напруги на вказаному контакті (у разі — контакті 8)

Закінчення табл. 9.2

Команда	Приклад	Опис
Цифрове введення/виведення		
pulseIn	i = pulseIn(8, HIGH);	Повертає тривалість у мікросекундах наступного імпульсу з напругою HIGH на контакті 8
tone	tone(8, 440, 1000);	Генерує на контакті серію 8 імпульсів з частотою 440 Гц тривалістю 1000 мс
noTone	noTone();	Перериває будь-які серії імпульсів, запущені викликом tone
Аналогове введення/виведення		
analogRead	int r; r = analogRead(0);	Надає змінній r значення в діапазоні від 0 до 1023. Значення 0 відповідає напрузі 0 на контакті 0, а значення 1023 - напрузі 5 В (або 3,3 В, якщо для живлення плати використовується напруга 3,3 В)
analogWrite	analogWrite(9, 127);	Виводить широтно-імпульсний сигнал. Протяжність позитивного імпульсу може змінюватися в діапазоні від 0 до 255, де число 255 відповідає 100%. Цю функцію можна передавати на номери контактів, позначених на платі як PWM (контакти 3, 5, 6, 9, 10 та 11)
Команди для роботи з часом		
millis	unsigned long l; l = millis();	Змінні типу long Arduino зберігають 32-бітові значення. Значення, що повертається функцією millis(), це число мілісекунд, що минули від моменту останнього скидання плати. Приблизно за 50 днів значення обнулюється і рахунок починається заново
micros	long l; l = micros();	Діє подібно до millis, але повертає число мікросекунд, що минули від моменту останнього скидання плати. Значення обнулюється приблизно кожні 70 хв
delay	delay(1000);	Припиняє роботу скетчу на 1000 мс, або на 1 с
delayMicroseconds	delayMicroseconds(100)	Припиняє роботу скетчу на 100 мкс. Зверніть увагу: мінімальна затримка становить 3 мкс, максимальна – близько 16 мс
Переривання		
attachInterrupt	attachInterrupt(1, myFunction, RISING);	Встановлює функцію myFunction як обробник позитивного фронту переривання 1 (контакт D3 в UNO)
detachInterrupt	detachInterrupt(1);	Забороняє обробку сигналів від переривання 1

Фреймворк Wiring для Arduino містить прості у використанні функції керування контактами на платі Arduino, а основна його частина написана на мові C. В каталозі Arduino IDE є директорія hardware, всередині якої – підкаталог arduino, в якому містяться пакети ядра Arduino. У внутрішній теці ядер міститься тека із файлами мовою C і розширенням .h та файлами реалізації на мові C plus з розширенням .cpp (рис. 9.17).

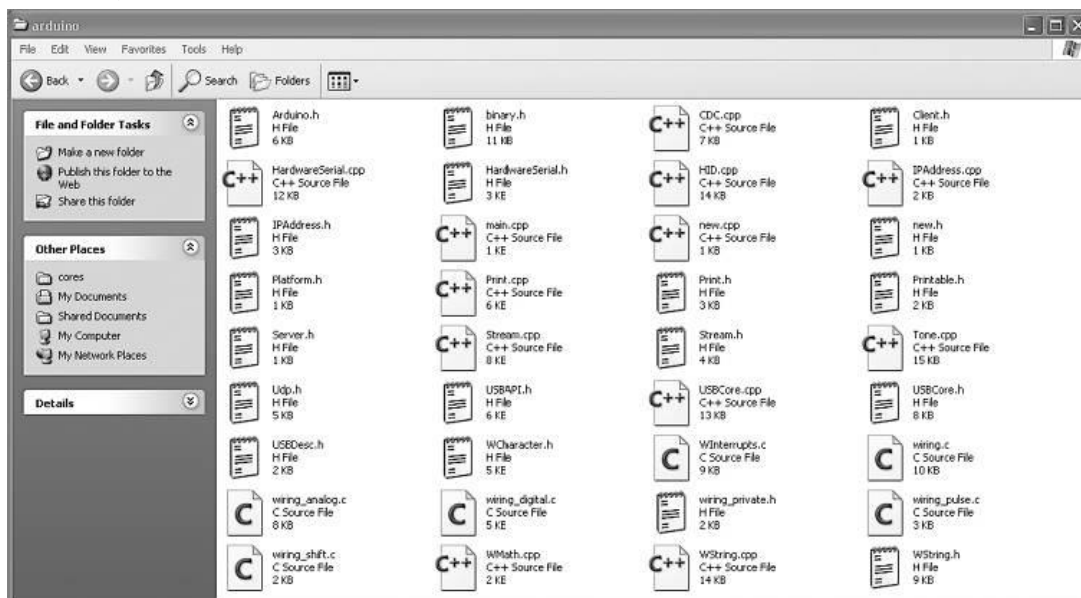


Рис. 9.17. Внутрішня структура теки cores

Файл Arduino.h складається з багатьох інструкцій #include, які містять визначення з інших файлів заголовків в теці cores/arduino і які в процесі компіляції будуть додані до основної програми (перетворення скетчу у форму, придатну для запису у флеш-пам'ять мікроконтролера). Наприклад, визначення констант матиме такий вигляд:

```
#define HIGH 0x1
#define LOW 0x0
#define INPUT 0x0
#define OUTPUT 0x1
#define INPUT_PULLUP 0x2
```

Директива define дозволяє дати ім'я константі перед тим, як програма буде скомпільована. Це означає, що звернувшись до імені HIGH, наприклад, програма отримає значення 1. Значення визначено, як 0x1, а не як 1, тому що в цьому файлі всі значення визначаються в шістнадцятковому форматі (у системі числення з основою 16). Коли скетч буде перетворений на формат, придатний для запису у флеш-пам'ять мікроконтролера, всі слова HIGH, LOW та інші автоматично

будуть перетворені на відповідні числа. Це дає певні переваги перед використанням змінних, оскільки не потрібно виділяти пам'ять для їхнього зберігання.

Також у файлі `arduino.h` є безліч сигнатур функцій, наприклад таких:

```
void pinMode(uint8_t, uint8_t);
void digitalWrite(uint8_t, uint8_t);
int digitalRead(uint8_t);
int analogRead(uint8_t);
void analogReference(uint8_t mode);
void analogWrite(uint8_t, int);
```

Зазначені записи попереджають компілятор про функції, які фактично реалізуються в іншому місці. Наприклад, перша сигнатура вказує, що функція `pinMode` приймає два аргументи (які представляють номер контакту і його режим) типу `uint8_t`. Ключове слово `void` вказує, що після запуску функція не повертає значення.

У діалекті мови C для Arduino тип `int` представляє цілі 16-бітові значення зі знаком в діапазоні між -32 768 і 32 767. Однак номер контакту не може бути негативним, і напевно чи на платі Arduino буде 32 767 контактів, тому для типу параметрів функції вибрано `uint8_t`. Тип `uint_8` означає, що це беззнаковий (`unsigned`) тип, який набуває цілих значень розміром 8 біт, тобто це числа в діапазоні між 0 і 255.

Також в теці Arduino можна знайти файл `main.cpp` з таким вмістом:

```
int main(void) {
    init();
    #if defined(USBCON)
        USBDevice.attach();
    #endif
    setup();
    for (;;) {
        loop();
        if (serialEventRun) serialEventRun();
    }
    return 0;
}
```

Функція `main()` автоматично буде запущена під час запуску програми. Це головна точка входу до програми. Це твердження справедливе і для програм Arduino. Розробникам скетчів на Arduino потрібно реалізувати лише дві функції – `setup` і `loop`, а функція `main` запустить `setup()` і потім перейде у нескінченний цикл `for`, де буде запускати функцію `loop()`. Команда `for(;;)` – це спосіб запису `while(true)` – нескінченного циклу. Крім запуску функції `loop()`, всередині циклу `for` є також команда `if`, яка перевіряє надходження даних у послідовний порт і

обслуговує їх. Функція `init()`, визначення якої знаходиться у файлі `wiring.c`, запускає обробку переривань. Наступні рядки

```
#if defined(USBCON)
  USBDevice.attach();
#endif
```

є ще однією директивою препроцесора C. Цей код діє подібно до команди `if`, але виконується не тоді, коли скетч вже працює в Arduino. Перевірка умови в директиві `#if` відбувається під час компіляції скетчу. Ця директива дозволяє вмикати та вимикати фрагменти коду залежно від конкретного типу плати. Якщо Arduino підтримує інтерфейс USB, у програму вмикається код, що під'єднує (ініціалізує) його, інакше компіляція не виконується.

Скетчі для Arduino зберігаються у вигляді текстових файлів з розширенням `.ino` у теці з таким самим ім'ям, але без розширення. Коли виконується завантаження скетча на плату, тоді спочатку препроцесор Arduino IDE збирає всі потрібні файли. Якщо в теці зі скетчем є інші файли, вони також будуть долучені до процесу збирання. Файли з програмним кодом мовою C компілюються окремо. На початку головного файла скетчу додається рядок, що містить `arduino.h`. Оскільки є безліч різних моделей плат Arduino, що використовують різні мікроконтролери з різними найменуваннями контактів, Arduino IDE має вибрати правильні визначення. В теці `hard-ware/arduino/variants` є окремі теки для всіх моделей плат Arduino, у кожній з яких зберігається файл `pins_arduino.h`. Цей файл містить імена контактів для заданої платформи. Після об'єднання файлів, запускається компілятор GCC, який приймає скетч, файл заголовків та файли реалізації з вихідним кодом на C і перетворює їх на код, який може виконувати мікроконтролер на платі Arduino. Компілятор виконує такі кроки:

1. Препроцесор інтерпретує всі команди `#if` і `#define` та визначає, який код має бути скомпільований.
2. Виконується компіляція і компонування коду в єдиний файл, придатний до виконання процесором на платі.
3. Коли компілятор завершить свою роботу, запускається відкритий програмний компонент з назвою `avrdude`, який пересилає в плату двійковий код, що виконується в шістнадцятковому форматі через послідовний інтерфейс USB.

На кожному мікроконтролері плат Arduino є невелика резидентна програма, яка називається завантажувачем (bootloader). Завантажувач виконується щоразу, коли відбувається скидання плати Arduino. Саме тому, коли відбувається передавання даних за послідовним інтерфейсом, апаратура зв'язку Arduino Uno виробляє примусове скидання плати, щоб дати завантажувачу можливість перевірити вхідні скетчі. Якщо отримано скетч, плата Arduino програмує сама себе, розпаковуючи шістнадцяткове представлення програми в двійкове та зберігаючи його у флеш-пам'яті. Коли наступного разу відбудеться скидання плати, після звичайної перевірки на наявність нового скетчу завантажувач автоматично запустить програму, що зберігається у флеш-пам'яті.

Контрольні запитання

1. Що таке мікроконтролер?
2. Що таке Arduino?
3. Наведіть основні характеристики Arduino.
4. Які види контактів є в Arduino?
5. Як програмується Arduino?
6. Що таке монітор порта?
7. Як керувати цифровими контактами в Arduino?
8. Як використовують аналогові контакти Arduino?
9. Як працює функція переривання Arduino?
10. Навіщо потрібна функція loop в Arduino?
11. Навіщо потрібна функція setup в Arduino?

Список літератури

1. *Алексієв В.О.* Мехатроніка транспортних засобів і систем : навч. посіб. / В.О. Алексієв, В.П. Волков, В.І. Калмиков. – Харків : ХНАДУ. – 2003. – 225 с.
2. *Ловейкін В.С.* Мехатроніка: підручник / В.С. Ловейкін, Ю.О. Ромасевич, В.В. Крушельницький. – Київ : КОМПРИНТ, 2020. – 404 с.
3. *Грабченко А.И.* Введение в мехатронику / А.И. Грабченко, В.Б. Клепиков, В.Л. Доброскок. – Харьков : НТУ «ХПИ», 2014. – 264 с.
4. *Артюх О.М.* Основи мехатроніки : навч. посіб. / О.М. Артюх, О.В. Дударенко, В.В. Кузьмін та ін. – Запоріжжя: НУ «Запорізька політехніка», 2021. – 372 с.
5. *William Bolton.* Mechatronics: electronic control systems in mechanical and electrical engineering. Harlow: Pearson, – 2015. – 663 p.
6. *David G. Alciatore.* Introduction to mechatronics and measurement systems. New York : McGraw-Hill Education, – 2018. – 609 p.
7. *Erika Ottaviano.* Mechatronics for cultural heritage and civil engineering. Cham: Springer International Publishing, – 2018. – 372 p.
8. *Bishop. Robert H.* The Mechatronics handbook / Robert H. Bishop. – Austin: The University of Texas at Austin. – 2002. – 1229 p.
9. *Сучасні перетворювачі частоти в системах електропривода :* навч. посіб. / М. В. Загірняк, Т. В. Коренькова, А. П. Калінов та ін. – 2-ге вид., переробл. і допов. – Харків : Точка, 2017. – 206 с.
10. *Опорний конспект лекцій з дисципліни «Теорія автоматичного управління» для студентів ОКР «бакалавр», 6050201 – «Системна інженерія» / уклад. : Я.М. Николайчук, Н.Я. Возна. – Тернопіль : Гал-друк, 2015. – 59 с.*
11. *Безвесільна О.М.* Елементи і пристрої автоматики та систем управління : навч. посіб. / О.М. Безвесільна, Т.О. Толочко. – Київ : КПІ ім. І. Сікорського, 2023. – 328 с.
12. *Бесекерский В.А.* Теория систем автоматического регулирования / В.А. Бесекерский, Е.П. Попов. – М. : Наука, 1972. – 302 с.
13. *Введение в мехатронику: учеб. пособие / А.И. Грабченко, В.Б. Клепиков, В.Л. Доброскок и др. – Харків : НТУ «ХПИ», 2014. – 264 с.*

14. *Пелевін Л.Є.* Гідравліка, гідромашини та гідропневмоавтоматика : підручник / Л.Є. Пелевін, Д.О. Міщук, В.П. Рашківський та ін. – Київ : КНУБА, МОНУ, 2015. – 340 с.
15. *Орловський Б.В.* Мехатроніка в галузевому машинобудуванні : навч. посіб. / Б. В. Орловський. – Київ : КНУТД, 2018. – 416 с.
16. *Пелевін Л.Є.* Механотронні системи гідропневмоавтоматики : навч. посіб. / Л.Є. Пелевін, М.М. Балака, Г.О. Аржаєв. – Київ : Аграр Медіа Груп, 2014. – 192 с.
17. *Гідравліка, гідропривод та гідро-і пневмоавтоматика : метод. вказівки до виконання лабораторних робіт / уклад. : Л.Є. Пелевін, Є.В. Горбатюк, С.Ю. Комоцька.* – Київ : КНУБА, 2019. – 36 с.

Навчальне видання

МІЩУК Дмитро Олександрович,
ВОЛЯНЮК Володимир Олександрович,
КОМОЦЬКА Світлана Юріївна

ЛОГІСТИКА МЕХАТРОННИХ СИСТЕМ БУДІВЕЛЬНОЇ ТЕХНІКИ

Конспект лекцій

Редагування і коректура *Г.В. Кобриної*
Комп'ютерне верстання *Т.І. Кукарєвої*

Підписано до друку 11.10. 2024. Формат 60 × 84_{1/16}
Ум. друк. арк 10,23. Обл.-вид. арк 11,0.
Електронний документ. Вид. № 20/І–24.

Видавець і виготовлювач
Київський національний університет будівництва і архітектури

Проспект Повітряних Сил, 31, Київ, Україна, 03037

Свідоцтво про внесення до Державного реєстру суб'єктів
видавничої справи ДК № 808 від 13.02.2002 р.

