

Міністерство освіти і науки України
Харківський національний університет імені В. Н. Каразіна

ПРОЄКТУВАННЯ ТА ТЕХНОЛОГІЇ КІБЕРФІЗИЧНИХ ВИРОБНИЧИХ СИСТЕМ

Конспект лекцій
для здобувачів вищої освіти другого (магістерського) рівня
денної та заочної форм здобуття освіти за спеціальністю
G7 «Автоматизація, комп'ютерно-інтегровані технології та робототехніка»

Електронний ресурс

Рецензенти:

О. О. Литвин – доктор фізико-математичних наук, професор, завідувач кафедри ХТЛПід Навчально-наукового інституту «Українська інженерно-педагогічна академія»;

А. М. Чернюк – кандидат технічних наук, професор, завідувач кафедри електроніки та електроенергетики Навчально-наукового інституту «Українська інженерно-педагогічна академія».

*Затверджено до розміщення в мережі Інтернет рішенням Науково-методичної ради
Харківського національного університету імені В. Н. Каразіна
(протокол № 1 від 23 жовтня 2025 року)*

П 79 **Проектування** та технології кіберфізичних виробничих систем : конспект лекцій для здобувачів вищої освіти другого (магістерського) рівня денної та заочної форм здобуття освіти за спеціальністю G7 «Автоматизація, комп'ютерно-інтегровані технології та робототехніка» [Електронний ресурс] / уклад. А. Ю. Мезеря. – Харків : ХНУ імені В. Н. Каразіна, 2026. – (PDF 191 с.)

Конспект лекцій з дисципліни «Проектування та технології кіберфізичних виробничих систем» розроблено на основі робочої програми. Розглянуто основні положення створення кіберфізичних систем та їхня експлуатація.

Наведено навчально-методичні матеріали для вивчення особливостей сучасної автоматизації технологічних процесів та виробництв. Розглянуто основні поняття кіберфізичних систем та цифровізації виробництв.

Конспект лекцій призначений здобувачам освітнього ступеня «магістр» денної та заочної форм навчання за спеціальністю G7 «Автоматизація, комп'ютерно-інтегровані технології та робототехніка».

УДК 658.5.011.56 : 658.589 (075.8)

© Харківський національний університет
імені В. Н. Каразіна, 2026

© Мезеря А. Ю., уклад., 2026

Зміст

ВСТУП.....	9
МЕТА ТА ЗАВДАННЯ КУРСУ	11
ЛЕКЦІЯ 1. КІБЕРФІЗИЧНІ ВИРОБНИЧІ СИСТЕМИ: ПОНЯТТЯ, СТРУКТУРА ТА ПРИНЦИПИ ПОБУДОВИ	14
1.1. Поняття кіберфізичної системи та кіберфізичної виробничої системи.....	14
1.2. Відмінності між автоматизованими, комп'ютерно-інтегрованими та кіберфізичними системами.....	16
1.3. Фізична, обчислювальна, комунікаційна та інформаційна складові	17
1.4. Взаємодія фізичних процесів і програмно-алгоритмічного середовища	18
1.5. Ієрархічна та розподілена організація виробничих систем.....	19
1.6. Основні властивості кіберфізичних виробничих систем	19
1.7. Обмеження та проблеми переходу до кіберфізичного виробництва.....	20
ЛЕКЦІЯ 2. АРХІТЕКТУРА КІБЕРФІЗИЧНИХ ВИРОБНИЧИХ СИСТЕМ.....	22
2.1. Принципи архітектурного проєктування	23
2.2. Функціональна декомпозиція виробничої системи	24
2.3. Рівні фізичних пристроїв, керування, периферійних обчислень і корпоративних сервісів	24
2.4. Централізовані, децентралізовані та розподілені архітектури ...	25
2.5. Сервісно-орієнтований підхід до побудови виробничих систем	26
2.6. Модульність, масштабованість та інтероперабельність.....	27
2.7. Архітектурні моделі RAMI 4.0 та Industrial Internet Reference Architecture.....	28
2.8. Вибір архітектури залежно від структури технологічного процесу	29
ЛЕКЦІЯ 3. МЕТОДИ ФОРМАЛІЗАЦІЇ ТА МОДЕЛЮВАННЯ КІБЕРФІЗИЧНИХ ВИРОБНИЧИХ СИСТЕМ.....	33
3.1. Модель як основа проєктування кіберфізичної системи.....	34
3.2. Моделі безперервних, дискретних та гібридних процесів.....	35
3.3. Моделі стану виробничих об'єктів	36
3.4. Диференціальні та різницеві моделі	37
3.5. Дискретно-подієве моделювання.....	38

3.6. Кінцеві автомати та мережі Петрі.....	38
3.7. Моделювання взаємодії фізичної та кібернетичної частин	39
3.8. Адекватність, точність та обчислювальна складність моделей..	40
ЛЕКЦІЯ 4. ІДЕНТИФІКАЦІЯ, СЕНСОРНІ СИСТЕМИ ТА ФОРМУВАННЯ ВИРОБНИЧИХ ДАНИХ	43
4.1. Структура вимірювального каналу кіберфізичної системи	44
4.2. Датчики та інтелектуальні сенсорні вузли.....	45
4.3. Аналогові та цифрові вимірювальні сигнали	46
4.4. Дискретизація та квантування сигналів	47
4.5. Похибки вимірювання та невизначеність даних	48
4.6. Фільтрація та первинне оброблення вимірювальної інформації	49
4.7. Ідентифікація параметрів виробничого об'єкта	49
4.8. Контекстуалізація виробничих даних	50
ЛЕКЦІЯ 5. ВИКОНАВЧІ ПРИСТРОЇ ТА ЛОКАЛЬНІ СИСТЕМИ АВТОМАТИЧНОГО КЕРУВАННЯ	53
5.1. Місце виконавчих пристроїв у структурі кіберфізичної системи	54
5.2. Електричні, пневматичні та гідравлічні виконавчі механізми ...	55
5.3. Регульований електропривод як компонент виробничої системи	57
5.4. Замкнені контури автоматичного регулювання	58
5.5. ПІ-, ПД- та цифрові регулятори.....	58
5.6. Дискретне керування технологічними операціями.....	59
5.7. Координація кількох локальних контурів.....	60
5.8. Обмеження виконавчих пристроїв та їх урахування в алгоритмах керування	61
ЛЕКЦІЯ 6. ПРОГРАМОВАНІ ЛОГІЧНІ КОНТРОЛЕРИ ТА СИСТЕМИ КЕРУВАННЯ РЕАЛЬНОГО ЧАСУ	64
6.1. Архітектура програмованого логічного контролера.....	65
6.2. Цикл виконання програми ПЛК.....	67
6.3. Стандарт ІЕС 61131-3 та мови програмування контролерів.....	67
6.4. Логічне, послідовне та регульовальне керування	69
6.5. Поняття системи реального часу	69
6.6. Детермінованість і часові обмеження	70
6.7. Час реакції системи та затримки оброблення.....	71
6.8. Розподілене керування виробничими об'єктами	72

ЛЕКЦІЯ 7. ПРОМИСЛОВІ МЕРЕЖІ ТА ТЕХНОЛОГІЇ ОБМІНУ ДАНИМИ.....	75
7.1. Комунікаційна підсистема кіберфізичного виробництва.....	75
7.2. Топології промислових мереж	76
7.3. Ethernet-технології у виробничих системах.....	77
7.4. Modbus TCP, PROFINET та EtherCAT	78
7.5. OPC UA як засіб інформаційної інтеграції	79
7.6. MQTT та обмін повідомленнями в IoT-системах	80
7.7. Затримка, джиттер, пропускна здатність і втрати пакетів	81
7.8. Синхронізація часу в розподіленій КФВС.....	82
7.9. Вибір мережевої технології для виробничого застосування	83
ЛЕКЦІЯ 8. ПРОМИСЛОВИЙ ІНТЕРНЕТ РЕЧЕЙ ТА ПЕРИФЕРІЙНІ ОБЧИСЛЕННЯ.....	85
8.1. Концепція Industrial Internet of Things	86
8.2. IoT-пристрої та шлюзи.....	87
8.3. Ідентифікація та адресація виробничих об'єктів	88
8.4. Edge computing у виробничих системах.....	89
8.5. Fog- та cloud-рівні оброблення даних	90
8.6. Розподіл обчислювальних задач між ПЛК, edge- і cloud-рівнями	91
8.7. Затримки передавання та оброблення інформації	92
8.8. Вибір місця виконання алгоритмів керування й аналітики	93
ЛЕКЦІЯ 9. ІНТЕГРАЦІЯ ВИРОБНИЧИХ СИСТЕМ: SCADA, MES, ERP ТА ІНФОРМАЦІЙНІ МОДЕЛІ.....	96
9.1. Вертикальна та горизонтальна інтеграція виробництва.....	97
9.2. SCADA-системи та диспетчерське керування	98
9.3. MES як рівень оперативного управління виробництвом	99
9.4. Взаємодія MES і ERP	100
9.5. ISA-95 та структуризація рівнів виробничого управління.....	101
9.6. Інформаційні моделі технологічних об'єктів	102
9.7. Семантична сумісність компонентів	102
9.8. Обмін технологічними, виробничими та корпоративними даними	103
ЛЕКЦІЯ 10. ЦИФРОВІ ДВІЙНИКИ КІБЕРФІЗИЧНИХ ВИРОБНИЧИХ СИСТЕМ.....	107

10.1. Поняття цифрової моделі, цифрової тіні та цифрового двійника	108
10.2. Структура цифрового двійника виробничого об'єкта	109
10.3. Синхронізація фізичного об'єкта та його цифрової моделі	110
10.4. Математичні, імітаційні та інформаційні моделі цифрового двійника.....	111
10.5. Оцінювання стану об'єкта за вимірювальними даними.....	112
10.6. Використання цифрових двійників для прогнозування поведінки системи.....	112
10.7. Віртуальне налагодження алгоритмів керування.....	113
10.8. Межі достовірності цифрового двійника.....	114
ЛЕКЦІЯ 11. ЗБИРАННЯ, ЗБЕРІГАННЯ ТА АНАЛІТИЧНЕ ОБРОБЛЕННЯ ВИРОБНИЧИХ ДАНИХ.....	117
11.1. Життєвий цикл виробничих даних	118
11.2. Потоківі та історичні дані	119
11.3. Часові ряди технологічних параметрів	120
11.4. Промислові бази даних та historians	121
11.5. Попередня обробка й очищення даних	122
11.6. Статистичні характеристики технологічних процесів	123
11.7. Кореляційний та регресійний аналіз	124
11.8. Виявлення аномальних станів.....	124
11.9. Використання даних для оцінювання якості та стану обладнання	125
ЛЕКЦІЯ 12. МАШИННЕ НАВЧАННЯ В КІБЕРФІЗИЧНИХ ВИРОБНИЧИХ СИСТЕМАХ.....	129
12.1. Місце машинного навчання у структурі КФВС	130
12.2. Формування навчальної та тестової вибірок	130
12.3. Ознаки та підготовка даних для машинного навчання.....	131
12.4. Навчання з учителем: регресія та класифікація	132
12.5. Навчання без учителя та кластеризація.....	133
12.6. Нейронні мережі та глибоке навчання	134
12.7. Навчання моделі та оптимізація її параметрів	135
12.8. Перенавчання та здатність моделі до узагальнення	136
12.9. Метрики класифікації та проблема незбалансованих даних ..	136
12.10. Інтерпретованість і межі використання машинного навчання	137

ЛЕКЦІЯ 13. КІБЕРБЕЗПЕКА КІБЕРФІЗИЧНИХ ВИРОБНИЧИХ СИСТЕМ	141
13.1. Особливості кібербезпеки операційних технологій	142
13.2. Основні властивості захищеної КФВС	143
13.3. Поверхня атаки та типові загрози	144
13.4. Оцінювання кіберризиків	145
13.5. Сегментація мережі, зони та канали зв'язку	146
13.6. Автентифікація, авторизація та віддалений доступ	147
13.7. Захист цілісності, оновлення, резервні копії та конфігурації	148
13.8. Моніторинг і виявлення аномальної поведінки	148
13.9. Реагування на кіберінциденти та відновлення	149
ЛЕКЦІЯ 14. БАГАТОАГЕНТНІ СИСТЕМИ ТА ДЕЦЕНТРАЛІЗОВАНЕ КЕРУВАННЯ КІБЕРФІЗИЧНИМ ВИРОБНИЦТВОМ	153
14.1. Централізоване та децентралізоване керування виробництвом	154
14.2. Поняття виробничого агента	155
14.3. Структура багатоагентної виробничої системи	156
14.4. Комунікація та координація агентів	157
14.5. Розподілене призначення виробничих завдань	158
14.6. Переговорні та аукціонні механізми	158
14.7. Самоорганізація та реконфігурація КФВС	159
14.8. Багатоагентне навчання з підкріпленням	160
14.9. Обмеження автономного та багатоагентного керування	161
ЛЕКЦІЯ 15. МЕТОДОЛОГІЯ ПРОЄКТУВАННЯ КІБЕРФІЗИЧНИХ ВИРОБНИЧИХ СИСТЕМ	165
15.1. Життєвий цикл проєктування КФВС	166
15.2. Аналіз технологічного об'єкта та формування вимог	167
15.3. Функціональна декомпозиція	168
15.4. Формування архітектури КФВС	169
15.5. Вибір сенсорної та виконавчої підсистем	170
15.6. Вибір контролерів та розподіл обчислень	171
15.7. Проєктування комунікаційної та інформаційної структури ...	172
15.8. Проєктування програмної архітектури	173
15.9. Модельно-орієнтоване проєктування та цифровий прототип	174

15.10. Верифікація, валідація та простежуваність вимог	174
ЛЕКЦІЯ 16. КІБЕРФІЗИЧНІ ВИРОБНИЧІ СИСТЕМИ В КОНЦЕПЦІЇ INDUSTRY 5.0	178
16.1. Перехід від Industry 4.0 до Industry 5.0.....	179
16.2. Людиноцентрична кіберфізична виробнича система	180
16.3. Людина в контурі прийняття рішень.....	181
16.4. Спільна робота людини та робота	182
16.5. Стале функціонування КФВС	182
16.6. Стійкість виробничої системи до порушень.....	183
16.7. Контрольована деградація замість повної відмови.....	185
16.8. Штучний інтелект і цифрові двійники в Industry 5.0.....	185
16.9. Багатокритеріальна оцінка КФВС	186
16.10. Обмеження переходу до Industry 5.0	187
Висновки до лекції 16	187

ВСТУП

Кіберфізичні виробничі системи формуються на стику автоматичного керування, інформаційних технологій, промислових комунікацій, обчислювальних систем і технологій оброблення даних. Їх принципова відмінність від традиційних автоматизованих систем полягає не лише у ширшому використанні цифрових засобів, а у тісному поєднанні фізичного технологічного процесу з його обчислювальним представленням. Стан обладнання, параметри технологічного процесу та результати виконання виробничих операцій постійно перетворюються на дані, які використовуються для керування, діагностування, прогнозування та коригування режимів роботи.

У типовій кіберфізичній виробничій системі фізичний об'єкт не розглядається ізольовано від засобів вимірювання, програмованих контролерів, промислових мереж, серверного обладнання, інформаційних моделей і програмних сервісів. Усі ці компоненти утворюють єдиний контур взаємодії. Зміна фізичного параметра реєструється сенсорною підсистемою, перетворюється на цифрові дані, передається каналами зв'язку, обробляється відповідним алгоритмом і може спричинити керувальну дію через виконавчий механізм. Одночасно ці самі дані можуть використовуватися на рівні диспетчеризації, виробничого планування, технічного обслуговування або аналітики.

Таке поєднання фізичних і цифрових процесів змінює сам підхід до проєктування систем автоматизації. Якщо для локальної системи керування часто достатньо визначити об'єкт, вибрати датчики, регулятор і виконавчий механізм, то для кіберфізичної системи цього недостатньо. Необхідно одночасно враховувати структуру інформаційних потоків, часові характеристики передавання даних, мережеву архітектуру, розміщення обчислювальних ресурсів, сумісність компонентів, точність моделей, можливість масштабування, вимоги до безпеки та поведінку системи за відмов окремих елементів.

Проєктування кіберфізичних виробничих систем тому є багаторівневою задачею. На фізичному рівні визначаються технологічні об'єкти, сенсори та виконавчі пристрої. На рівні керування формуються локальні алгоритми регулювання, логічного та послідовного керування. На комунікаційному рівні проєктуються мережі, протоколи і механізми синхронізації. На інформаційному рівні задаються структури даних, моделі виробничих об'єктів та способи обміну інформацією між підсистемами.

Вищі рівні охоплюють диспетчеризацію, виробниче планування, аналітику, цифрові двійники та прогнозування технічного стану.

Окрема проблема полягає у часовій узгодженості цих рівнів. Для виробничої системи недостатньо лише передати правильні дані: у багатьох задачах вони мають бути отримані та оброблені в заданому часовому інтервалі. Затримка в мережі, перевантаження обчислювального вузла або несинхронність окремих компонентів можуть впливати не лише на інформаційний обмін, а й безпосередньо на якість керування фізичним процесом. Через це поняття реального часу, детермінованості, затримки та джиттера є невіддільними від аналізу кіберфізичних виробничих систем.

Ще один аспект пов'язаний із моделями. Кіберфізична система працює не лише з вимірними значеннями, а й з формалізованим описом об'єкта. Математична, імітаційна або інформаційна модель дозволяє оцінювати стан системи, перевіряти алгоритми, прогнозувати поведінку та виявляти відхилення. На цій основі формуються цифрові двійники, які поєднують модель фізичного об'єкта з даними, що надходять під час його реальної експлуатації. Достовірність такого двійника визначається не складністю моделі як такої, а тим, наскільки точно вона відтворює властивості об'єкта в межах поставленої задачі.

Використання методів аналізу даних і машинного навчання розширює можливості кіберфізичних систем, але не скасовує потреби у фізичному розумінні процесу. Модель, навчена на виробничих даних, може виявляти закономірності, прогнозувати параметри або класифікувати стани обладнання. Водночас її застосування обмежується якістю вихідних даних, діапазоном режимів, представлених у вибірці, і стабільністю технологічного процесу. Тому в інженерних задачах доцільно розглядати не протиставлення фізичних моделей і моделей, отриманих із даних, а можливість їх узгодженого використання.

Проектування кіберфізичної виробничої системи також не може обмежуватися штатним режимом. Система повинна бути проаналізована за умов відмов датчиків, порушення зв'язку, некоректних даних, перевантаження мережі, відмов виконавчих пристроїв і несанкціонованого втручання. Це безпосередньо пов'язує проектування з питаннями надійності, функціональної безпеки, діагностування та кібербезпеки. Рішення, прийняті на етапі побудови архітектури, часто визначають межі можливого резервування, локалізації несправностей і подальшого масштабування системи.

Зміст курсу побудовано від базових принципів до повного циклу проектування. Спочатку розглядаються структура, архітектура та

математичний опис кіберфізичних виробничих систем. Далі аналізуються сенсорні та виконавчі підсистеми, програмовані логічні контролери, технології реального часу і промислові мережі. Наступний блок присвячено Industrial Internet of Things, периферійним обчисленням, SCADA, MES, ERP та цифровим двійникам. Після цього розглядаються методи роботи з виробничими даними, елементи машинного навчання, надійність, функціональна безпека і кіберзахист. Завершальні лекції поєднують окремі компоненти у послідовну методологію проектування, верифікації, валідації та введення системи в експлуатацію.

Такий порядок дозволяє розглядати кіберфізичну виробничу систему не як набір окремих технологій, а як узгоджений інженерний об'єкт, у якому рішення на одному рівні впливають на властивості інших рівнів. Саме ця взаємозалежність є основою подальшого викладення матеріалу.

МЕТА ТА ЗАВДАННЯ КУРСУ

Метою курсу лекцій є формування цілісного розуміння принципів побудови, моделювання, проектування та функціонування кіберфізичних виробничих систем, а також набуття здатності обґрунтовано визначати їх архітектуру, склад технічних і програмних засобів, структуру інформаційного обміну та алгоритми взаємодії між фізичними й цифровими компонентами.

Курс орієнтований не на вивчення окремих програмних або апаратних платформ, а на опанування загальних інженерних принципів, які залишаються придатними за зміни конкретних контролерів, мережевих протоколів, обчислювальних платформ або програмних середовищ. Основним об'єктом аналізу виступає система як цілісність: від вимірювання фізичного параметра до формування керувальної дії, передавання даних, їх оброблення, зберігання та використання на вищих рівнях виробничого управління.

Для досягнення поставленої мети курс передбачає розв'язання таких завдань:

- сформувати понятійний апарат, необхідний для опису кіберфізичних виробничих систем, та встановити відмінності між традиційною автоматизацією, комп'ютерно-інтегрованими системами і кіберфізичними системами;
- розглянути принципи структурної та функціональної декомпозиції виробничого об'єкта, виділення фізичного, керувального, комунікаційного, інформаційного та обчислювального рівнів;

- опанувати основні підходи до архітектурного проєктування централізованих, розподілених і сервісно-орієнтованих виробничих систем;
- засвоїти методи математичного опису безперервних, дискретних і гібридних процесів, а також способи подання станів і подій у виробничих системах;
- розглянути принципи побудови сенсорних підсистем, формування вимірювальних даних, дискретизації, фільтрації та оцінювання похибок;
- вивчити структуру виконавчих підсистем і локальних контурів автоматичного керування, а також особливості взаємодії регуляторів із фізичними об'єктами;
- сформуванати розуміння принципів роботи програмованих логічних контролерів, циклічного виконання програм та організації керування в реальному часі;
- дослідити характеристики промислових мереж і протоколів, що визначають можливість передавання технологічних даних із заданими часовими характеристиками;
- встановити умови застосування технологій Industrial Internet of Things, edge computing, fog computing та cloud computing у виробничих системах;
- розглянути принципи вертикальної та горизонтальної інтеграції виробничих підсистем, а також взаємодію між рівнями PLC, SCADA, MES та ERP;
- сформуванати навички побудови інформаційних і математичних моделей виробничих об'єктів та їх використання у складі цифрових двійників;
- опанувати базові методи аналізу часових рядів, статистичного оброблення виробничих даних та виявлення відхилень у роботі обладнання;
- розглянути можливості та обмеження застосування методів машинного навчання для прогнозування параметрів, класифікації станів і виявлення аномалій;
- сформуванати уявлення про методи оцінювання надійності, резервування, діагностування та забезпечення відмовостійкої роботи кіберфізичних систем;
- визначити основні загрози для промислових інформаційно-керувальних систем та принципи врахування вимог кібербезпеки на етапі проєктування;
- опанувати послідовність розроблення кіберфізичної виробничої системи — від постановки вимог і вибору архітектури до інтеграційного тестування, верифікації, валідації та введення в експлуатацію;

- сформувати здатність оцінювати проєктні рішення не за одним параметром, а за сукупністю критеріїв, серед яких точність, швидкодія, надійність, масштабованість, сумісність, вартість реалізації та можливість подальшої модернізації.

Після опрацювання курсу здобувач освіти має розуміти логіку функціонування кіберфізичної виробничої системи на всіх основних рівнях, уміти виділяти її компоненти та зв'язки, формувати вимоги до архітектури, аналізувати інформаційні й керувальні потоки, оцінювати часові та надійнісні характеристики, обирати технології обміну даними та обґрунтовувати структуру системи відповідно до властивостей конкретного виробничого об'єкта.

Результатом вивчення дисципліни має бути здатність перейти від опису технологічного процесу до обґрунтованої структури кіберфізичної системи, у якій фізичні, програмні, інформаційні та комунікаційні компоненти розглядаються як взаємопов'язані частини одного проєктного рішення.

ЛЕКЦІЯ 1. КІБЕРФІЗИЧНІ ВИРОБНИЧІ СИСТЕМИ: ПОНЯТТЯ, СТРУКТУРА ТА ПРИНЦИПИ ПОБУДОВИ

Мета лекції

Мета лекції полягає у формуванні системного уявлення про кіберфізичні виробничі системи, їх структуру, принципи побудови та особливості взаємодії фізичних і цифрових компонентів. Розглядаються відмінності між традиційними автоматизованими системами, комп'ютерно-інтегрованим виробництвом і кіберфізичними виробничими системами, а також питання розподілу функцій керування, оброблення даних і комунікацій між різними рівнями виробничої системи.

1.1. Поняття кіберфізичної системи та кіберфізичної виробничої системи

Кіберфізична система — Cyber-Physical System, CPS — це система, в якій фізичний процес безпосередньо пов'язаний з обчисленнями, передаванням даних та формуванням керувальних дій. Для промислових застосувань використовується поняття Cyber-Physical Production System, або кіберфізична виробнича система — КФВС [1–3].

КФВС не є окремим видом обладнання. Її не можна визначити лише за наявністю програмованого логічного контролера, промислової мережі, SCADA-системи або хмарної платформи. Визначальною є організація взаємодії між фізичним процесом та його цифровим представленням.

У типовій КФВС фізичний параметр спочатку вимірюється сенсором. Отриманий сигнал перетворюється у цифрові дані, передається до обчислювального пристрою та обробляється відповідно до заданого алгоритму. Результатом може бути керувальна команда, зміна режиму роботи, діагностичне повідомлення або прогноз подальшого стану обладнання.

Таким чином формується замкнений цикл:

фізичний процес → вимірювання → цифрові дані → оброблення → рішення → керувальна дія → фізичний процес.

Фізичний об'єкт у загальному випадку можна подати через рівняння стану:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), \mathbf{d}(t), t),$$

де $\mathbf{x}(t)$ — вектор змінних стану об'єкта; $\mathbf{u}(t)$ — вектор керувальних впливів; $\mathbf{d}(t)$ — вектор зовнішніх збурень; $\mathbf{f}$ — функція, що описує динаміку об'єкта; t — час.



Рисунок – Структура кіберфізичної системи

Рис. 1.1 – Структура кіберфізичної системи.

До змінних стану можуть належати температура, тиск, швидкість, положення механізму, рівень, витрата, момент, струм електродвигуна та інші фізичні величини.

Обчислювальна частина системи, однак, зазвичай не має інформації про всі внутрішні координати стану. Вона працює з вимірюваними величинами:

$$\mathbf{y}(t) = \mathbf{h}(\mathbf{x}(t), t) + \mathbf{v}(t),$$

де $\mathbf{y}(t)$ — вектор вимірюваних величин; $\mathbf{h}$ — функція вимірювання; $\mathbf{v}(t)$ — вектор похибок і шумів вимірювання.

Звідси випливає принципове обмеження: цифрова система приймає рішення не на основі безпосереднього доступу до фізичного процесу, а за його вимірюваним представленням. Неточність сенсорів, пропуски даних або затримки передавання впливають на подальше керування.

Кіберфізичним виробничим об'єктом може бути окремий електропривод, верстат, промисловий робот, насосна установка, технологічна комірка, транспортний модуль або ціла виробнича лінія. Масштаб системи не є визначальною ознакою. Значення має наявність двосторонньої взаємодії між фізичною та цифровою частинами [1, 4].

1.2. Відмінності між автоматизованими, комп'ютерно-інтегрованими та кіберфізичними системами

Кіберфізичні системи не замінюють класичну автоматизацію, а розширюють її можливості. Базовий контур автоматичного керування залишається необхідним і в КФВС.

У традиційній автоматизованій системі датчик вимірює параметр, регулятор порівнює його із заданим значенням, після чого виконавчий механізм впливає на технологічний об'єкт.

Похибка регулювання визначається як:

$$e(t) = r(t) - y(t),$$

де $r(t)$ — задане значення параметра; $y(t)$ — фактичне виміряне значення.

Для пропорційного закону керування:

$$u(t) = K_p e(t),$$

де $u(t)$ — керувальний вплив; K_p — коефіцієнт підсилення пропорційного регулятора;

Такий контур може повністю працювати без участі людини, але це ще не означає, що він є кіберфізичним. Наприклад, локальний регулятор температури може не обмінюватися інформацією з іншими пристроями, не формувати цифрову модель стану обладнання і не використовувати дані вищого рівня.

Комп'ютерно-інтегроване виробництво передбачає ширше об'єднання підсистем. Інформаційно пов'язуються автоматизоване проєктування, технологічна підготовка виробництва, обладнання, контроль якості, диспетчеризація та планування.

КФВС розвиває таку інтеграцію за рахунок постійного використання актуальної інформації про фізичний стан об'єктів, розподілених обчислень, цифрових моделей і можливості адаптації до зміни виробничих умов [2, 4, 5].

Наприклад, звичайний автоматизований верстат виконує задану програму. Комп'ютерно-інтегрований верстат додатково отримує завдання від інформаційної системи і передає результати його виконання. Кіберфізичний виробничий модуль може, крім цього, оцінювати навантаження, стан інструмента, температуру, енергоспоживання та інші параметри, а отримані дані використовувати для коригування режиму або прогнозування технічного стану.

Тому відмінність КФВС визначається не кількістю цифрових пристроїв, а способом їх взаємодії з фізичним процесом.

1.3. Фізична, обчислювальна, комунікаційна та інформаційна складові

У структурі КФВС доцільно виділяти фізичну, обчислювальну, комунікаційну та інформаційну складові [2, 3].

Фізична складова включає технологічне обладнання, механізми, електроприводи, роботів, конвеєри, клапани, насоси, інструмент та оброблюваний матеріал. До неї належать процеси механічного руху, нагрівання, охолодження, переміщення, змішування, обробки та інші фізичні перетворення.

На межі між фізичним і цифровим середовищем знаходяться сенсори та виконавчі пристрої. Сенсор перетворює фізичну величину на сигнал, придатний для цифрового оброблення. Виконавчий механізм здійснює зворотне перетворення — цифрова команда призводить до фізичної дії.

Обчислювальна складова охоплює ПЛК, промислові комп'ютери, контролери руху, мікроконтролери, edge-пристрої, сервери та інші засоби, які виконують програмні алгоритми.

У цифровій системі неперервний сигнал представляється послідовністю відліків:

$$x[k] = x(kT_d),$$

де $x[k]$ — значення сигналу в k -й момент дискретного часу; k — номер відліку; T_d — період дискретизації.

Частота дискретизації визначається як:

$$f_d = \frac{1}{T_d}.$$

Вибір періоду дискретизації залежить від швидкості зміни фізичного процесу. Для повільної температурної системи допустимий значно більший період, ніж для високодинамічного електропривода.

Комунікаційна складова забезпечує передавання вимірювань, керувальних команд, діагностичних повідомлень, виробничих завдань та іншої інформації між компонентами.

Орієнтовну швидкість формування даних від однакових вимірювальних каналів можна визначити як:

$$R = Nbf_d,$$

де N — кількість каналів; b — кількість бітів в одному відліку;

Фактичне навантаження мережі є вищим через службові дані протоколів, адресацію, контроль цілісності та інші накладні витрати.

Інформаційна складова визначає зміст даних. Числове значення без контексту не може бути правильно використане автоматично. Система

повинна знати, до якого обладнання належить параметр, яку фізичну величину він характеризує, в яких одиницях вимірюється і коли був отриманий.

Тому інформаційна модель містить не лише числа, а й ідентифікатори обладнання, назви параметрів, одиниці вимірювання, часові мітки, стани та зв'язки між компонентами.

1.4. Взаємодія фізичних процесів і програмно-алгоритмічного середовища

Фізичні та цифрові процеси у КФВС розвиваються взаємопов'язано. Наприклад, датчик виявляє деталь на конвеєрі. Контролер отримує інформацію, визначає стан наступної технологічної позиції і залежно від нього вмикає або зупиняє привід.

Керувальний вплив у загальному вигляді можна представити як:

$$\mathbf{u}[k] = \pi(\hat{\mathbf{x}}[k], \mathbf{r}[k]),$$

де π — алгоритм керування; $\hat{\mathbf{x}}[k]$ — оцінений стан об'єкта; $\mathbf{r}[k]$ — вектор заданих параметрів.

Алгоритм часто працює не з реальним повним станом об'єкта, а з його оцінкою. Тому достовірність рішення залежить від якості вимірювань і моделі.

Інша характеристика — час проходження інформації через повний контур.

$$\tau_{\Sigma} = \tau_{\text{вим}} + \tau_{\text{пер}} + \tau_{\text{обч}} + \tau_{\text{ком}} + \tau_{\text{вик}},$$

де τ_{Σ} — сумарна затримка; $\tau_{\text{вим}}$ — час вимірювання; $\tau_{\text{пер}}$ — час передавання даних; $\tau_{\text{обч}}$ — час оброблення; $\tau_{\text{ком}}$ — час передавання команди; $\tau_{\text{вик}}$ — затримка виконавчого механізму.

Для забезпечення заданої швидкодії має виконуватися умова:

$$\tau_{\Sigma} \leq T_{\text{доп}},$$

де $T_{\text{доп}}$ — максимально допустимий час реакції.

Отже, висока швидкість мережі сама по собі не гарантує швидкодії КФВС. Якщо обчислення займає надто багато часу або виконавчий механізм реагує повільно, мережевий канал не може компенсувати цю затримку.

Різні виробничі функції мають різні часові обмеження. Аварійне відключення та регулювання електропривода повинні виконуватися локально. Аналітика продуктивності або прогнозування ресурсу можуть виконуватися на серверному чи хмарному рівні.

1.5. Ієрархічна та розподілена організація виробничих систем

Класична автоматизація переважно має ієрархічну структуру. На нижньому рівні розташовані датчики й виконавчі пристрої, вище — ПЛК, далі — SCADA, системи виробничого управління та корпоративні інформаційні засоби.

Перевага такої структури полягає у чіткому розподілі функцій. Недолік проявляється при частих змінах виробничої конфігурації, оскільки значна частина логіки залежить від центрального рівня.

У КФВС частина функцій може передаватися локальним модулям. Виробнича комірka може самостійно контролювати власний стан, формувати діагностичні повідомлення, змінювати допустимий режим роботи та взаємодіяти з іншими компонентами [4].

Проте децентралізація потребує координації. Якщо два автономні модулі одночасно потребують одного ресурсу, виникає конфлікт. Для його вирішення використовують диспетчерські механізми, правила пріоритетів, сервісні або багатоагентні підходи.

На практиці найбільш доцільною є гібридна організація. Швидкі й критичні функції залишаються локальними, координація виконується на рівні установок або ліній, а довгострокове планування й аналітика — на верхніх рівнях.

1.6. Основні властивості кіберфізичних виробничих систем

До характерних властивостей КФВС відносять спостережуваність, зв'язаність, інтеперабельність, адаптивність, реконфігурованість, масштабованість, автономність та діагностованість [1–4].

Спостережуваність означає, що система має достатньо інформації для оцінювання стану фізичного об'єкта. Не всі параметри повинні вимірюватися безпосередньо, частина з них може визначатися за моделлю.

Зв'язаність характеризує можливість інформаційної взаємодії між компонентами. Проте надмірна кількість зв'язків збільшує складність і кількість потенційних відмов.

Інтеперабельність означає здатність різних пристроїв коректно обмінюватися та інтерпретувати інформацію. Для цього недостатньо лише однакового фізичного інтерфейсу; необхідно узгоджувати структуру даних, протокол, одиниці вимірювання та семантику.

Адаптивність полягає у зміні параметрів роботи відповідно до поточного стану системи. Наприклад, при підвищенні температури привода може автоматично знижуватися допустиме навантаження.

Реконфігурованість характеризує можливість змінювати склад або структуру системи без її повного перепроєктування.

Масштабованість визначає здатність архітектури працювати при збільшенні кількості компонентів, сигналів і інформаційних потоків.

Автономність означає можливість окремого модуля виконувати частину функцій без постійної команди центральної системи, але лише в межах заданих обмежень.

Діагностованість пов'язана з можливістю виявлення несправностей та оцінювання технічного стану за доступними вимірюваннями.

1.7. Обмеження та проблеми переходу до кіберфізичного виробництва

Однією з основних проблем є сумісність обладнання різних поколінь. Промислові машини експлуатуються значно довше, ніж інформаційні технології. Тому на підприємстві можуть одночасно працювати сучасні ПЛК, старі контролери та обладнання без цифрових інтерфейсів. Для їх інтеграції необхідні додаткові шлюзи та перетворювачі [2, 3].

Другим обмеженням є якість даних. Якщо вимірювання неправильне, цифрова модель також буде містити похибку.

Абсолютна похибка вимірювання визначається як:

$$\Delta x = x_{\text{вим}} - x_{\text{іст}},$$

де $x_{\text{вим}}$ — виміряне значення; $x_{\text{іст}}$ — істинне значення величини.

У реальних метрологічних задачах істинне значення точно невідоме, тому використовують оцінки похибок і невизначеності. Проте наведене співвідношення показує сам принцип виникнення відхилення.

При великій кількості компонентів швидко зростає потенційна складність зв'язків.

$$N_{\text{зв}} = \frac{n(n-1)}{2},$$

де $N_{\text{зв}}$ — максимальна кількість парних зв'язків; n — кількість компонентів.

Для 10 компонентів можливі 45 парних зв'язків, а для 100 — уже 4950. Саме тому великі КФВС потребують структурованої архітектури та стандартизованих інтерфейсів.

Крім цього, необхідно враховувати залежність від мережевого зв'язку. Для кожної функції має бути визначено поведінку у разі відмови каналу: продовження локальної роботи, перехід до резервного алгоритму, контрольоване зупинення або безпечний стан.

Окремою проблемою є кібербезпека. У КФВС атака на інформаційну частину може спричинити фізичні наслідки, оскільки спотворені дані або команди впливають на реальне обладнання.

Технічна можливість реалізувати певну цифрову функцію також не означає її доцільності. Витрати на датчики, мережі, програмне забезпечення, інтеграцію та підтримку повинні відповідати результату, який отримує виробнича система.

Висновки до лекції 1

Кіберфізична виробнича система являє собою поєднання фізичних об'єктів, сенсорів, виконавчих пристроїв, засобів обчислення, мереж та інформаційних моделей. Її принциповою ознакою є двосторонній зв'язок між фізичним процесом і цифровим середовищем.

КФВС не замінює традиційну автоматизацію. Локальні контури регулювання залишаються основою керування швидкими процесами, а кіберфізичний рівень розширює можливості координації, аналізу даних, діагностування та адаптації.

Фізична, обчислювальна, комунікаційна та інформаційна частини системи мають проектуватися узгоджено. Неточність сенсора, мережева затримка або помилка інформаційної моделі безпосередньо впливають на результат керування.

Розподілена архітектура підвищує гнучкість системи, але вимагає координації, стандартизації інтерфейсів і визначення поведінки окремих компонентів при порушенні зв'язку.

Контрольні питання

Що розуміють під кіберфізичною виробничою системою?

Чим КФВС відрізняється від традиційної автоматизованої системи?

Які складові входять до структури КФВС?

Яку функцію виконують сенсори та виконавчі механізми?

Як пов'язані частота і період дискретизації?

Чому затримки передавання даних впливають на керування?

У чому полягає різниця між ієрархічною та розподіленою структурами?

Що розуміють під інтероперабельністю?

Які властивості характеризують КФВС?

Які обмеження виникають під час переходу до кіберфізичного виробництва?

Список використаних джерел

1. Napoleone A., Macchi M., Pozzetti A. A review on the characteristics of cyber-physical systems for the future smart factories. *Journal of Manufacturing Systems*. 2020. Vol. 54. P. 305–335. DOI: 10.1016/j.jmsy.2020.01.007.
2. Pivoto D. G. S., de Almeida L. F. F., Righi R. R., Rodrigues J. J. P. C., Lugli A. B., Alberti A. M. Cyber-physical systems architectures for industrial internet of things applications in Industry 4.0: A literature review. *Journal of Manufacturing Systems*. 2021. Vol. 58. P. 176–192. DOI: 10.1016/j.jmsy.2020.11.017.
3. Dafflon B., Moalla N., Ouzrout Y. The challenges, approaches, and used techniques of CPS for manufacturing in Industry 4.0: a literature review. *The International Journal of Advanced Manufacturing Technology*. 2021. Vol. 113. P. 2395–2412. DOI: 10.1007/s00170-020-06572-4.
4. Suvarna M., Yap K. S., Yang W., Li J., Ng Y. T., Wang X. Cyber-Physical Production Systems for Data-Driven, Decentralized, and Secure Manufacturing—A Perspective. *Engineering*. 2021. Vol. 7, No. 9. P. 1212–1223. DOI: 10.1016/j.eng.2021.04.021.
5. Wang B., Tao F., Fang X., Liu C., Liu Y., Freiheit T. Smart Manufacturing and Intelligent Manufacturing: A Comparative Review. *Engineering*. 2021. Vol. 7, No. 6. P. 738–757. DOI: 10.1016/j.eng.2020.07.017.

ЛЕКЦІЯ 2. АРХІТЕКТУРА КІБЕРФІЗИЧНИХ ВИРОБНИЧИХ СИСТЕМ

Мета лекції

Мета лекції полягає у формуванні системного розуміння архітектури кіберфізичних виробничих систем, принципів їх функціональної декомпозиції та розподілу обчислювальних, керувальних і інформаційних функцій між окремими рівнями. Розглядаються централізовані, децентралізовані та розподілені архітектури, сервісно-орієнтований підхід, принципи модульності, масштабованості та інтероперабельності, а також референсні архітектурні моделі RAMI 4.0 та IIRA.

Архітектура КФВС визначає не тільки фізичне розміщення технічних засобів. Вона задає склад компонентів, функції кожного з них, характер взаємодії, правила обміну даними, місце виконання алгоритмів і межі відповідальності окремих підсистем. Помилка на цьому рівні зазвичай не усувається простим удосконаленням програмного алгоритму, оскільки її причиною може бути неправильний розподіл функцій, надмірна залежність від мережі або невдале розміщення критичних обчислень.

2.1. Принципи архітектурного проєктування

Архітектуру кіберфізичної виробничої системи доцільно розглядати як формалізований опис компонентів, їх функцій, інтерфейсів та зв'язків. Вона відповідає на декілька взаємопов'язаних питань: які компоненти входять до системи, що саме вони виконують, якою інформацією обмінюються, де здійснюється оброблення даних і яким чином локальні рішення узгоджуються із загальними виробничими цілями [1–3].

Структуру КФВС можна представити орієнтованим графом:

$$\mathcal{G} = (\mathcal{V}, \mathcal{E}),$$

де $\mathcal{G}$ — граф структури системи; $\mathcal{V}$ — множина компонентів; $\mathcal{E}$ — множина спрямованих зв'язків між компонентами.

Вершинами такого графа можуть бути датчики, контролери, приводи, промислові комп'ютери, сервери та програмні сервіси. Ребра описують інформаційні або керувальні зв'язки.

Наявність зв'язку між двома компонентами можна описати елементом матриці суміжності:

$$a_{ij} = \begin{cases} 1, & \text{якщо існує зв'язок від компонента } i \text{ до компонента } j, \\ 0, & \text{якщо такий зв'язок відсутній.} \end{cases}$$

де a_{ij} — елемент матриці суміжності; i — номер компонента-відправника; j — номер компонента-одержувача.

Таке представлення корисне не тільки для опису мережевої топології. Воно дозволяє аналізувати залежності між підсистемами. Якщо відмова одного вузла розриває всі шляхи між фізичним об'єктом та системою керування, архітектура містить критичну одиничну точку відмови.

Перший принцип архітектурного проєктування — відокремлення функції від конкретної реалізації. Спочатку визначають, що повинна робити система, а вже після цього вибирають контролер, мережу або програмну платформу. Зворотний підхід, коли функції підлаштовують під уже обране обладнання, часто створює зайві обмеження.

Другий принцип — локалізація критичних функцій. Керування, від якого залежить безпечний або стійкий стан фізичного процесу, не повинно без достатнього обґрунтування залежати від віддаленого сервера або нестабільного каналу зв'язку.

Третій принцип — явне визначення інтерфейсів. Кожна взаємодія повинна мати визначені вхідні й вихідні дані, формат повідомлення, часові вимоги та поведінку при відмові. Саме інтерфейси, а не окремі компоненти, часто стають основним джерелом проблем при інтеграції складної КФВС [1, 4].

2.2. Функціональна декомпозиція виробничої системи

Функціональна декомпозиція полягає у поділі загальної задачі на підзадачі, для кожної з яких можна визначити вхідні дані, вихідні результати, часові вимоги та відповідальний компонент.

Для типової КФВС можна виділити функції вимірювання, локального керування, координації, оброблення даних, діагностування, зберігання інформації, візуалізації, виробничого планування та інтеграції із зовнішніми системами.

Такий поділ не означає, що кожній функції обов'язково відповідає окремий фізичний пристрій. Один ПЛК може одночасно виконувати логічне керування, регулювання, локальну діагностику та попереднє оброблення даних. Навпаки, складна функція може бути розподілена між декількома компонентами.

Під час декомпозиції бажано уникати двох крайнощів. Надто великі функціональні блоки складно змінювати та тестувати. Надто дрібний поділ збільшує кількість інтерфейсів і робить поведінку системи залежною від великої кількості обмінів.

Функціональний блок доцільно виділяти окремо, якщо він має хоча б одну з таких властивостей: власний часовий режим, незалежні вхідні та вихідні дані, можливість автономного тестування або потребу у заміні без перебудови всієї системи.

Наприклад, у роботизованій комірці можна розділити функції визначення наявності деталі, керування роботом, контролю захисної зони, координації з конвеєром і передачі виробничих даних. Функція безпеки при цьому повинна залишатися працездатною незалежно від роботи системи статистики або виробничої аналітики.

Такий принцип називають *separation of concerns* — розділенням відповідальності. Він широко використовується в архітектурах CPPS, зокрема при проектуванні реконфігурованих виробничих систем [4, 5].

2.3. Рівні фізичних пристроїв, керування, периферійних обчислень і корпоративних сервісів

У практичній КФВС функції розподіляються між декількома рівнями. Точна кількість рівнів залежить від конкретної архітектури, проте для проектування зручно розглядати чотири основні групи: фізичні пристрої, локальне керування, периферійні обчислення та верхній інформаційний рівень.

Рівень фізичних пристроїв включає сенсори, виконавчі механізми, електроприводи, роботи, технологічне обладнання та інші компоненти, безпосередньо пов'язані з фізичним процесом.

Рівень керування формують ПЛК, контролери руху, регулятори та інші засоби, які виконують часово-критичні алгоритми. Саме тут доцільно залишати швидкі замкнені контури.

Периферійний рівень, або edge-рівень, розташовується між технологічним обладнанням і централізованими інформаційними системами. Тут можуть виконуватися агрегація даних, фільтрація, локальна аналітика, перетворення протоколів та частина діагностичних алгоритмів.

Верхній рівень включає сервери, системи диспетчеризації, MES, корпоративні сервіси та, за потреби, хмарні ресурси. Для нього характерні задачі, що працюють із більшими часовими інтервалами та значними масивами даних.

Розподіл функцій має визначатися не престижністю певної технології, а часовими і функціональними вимогами. Якщо повний маршрут передавання та оброблення інформації містить s послідовних етапів, його затримка становитиме:

$$\tau_{\text{марш}} = \sum_{q=1}^s \tau_q,$$

де $\tau_{\text{марш}}$ — сумарна затримка маршруту; s — кількість послідовних етапів; q — номер етапу; τ_q — затримка на відповідному етапі.

Якщо алгоритм регулювання допускає лише малу затримку, додавання віддаленого сервера у контур може зробити архітектуру непридатною незалежно від продуктивності цього сервера.

Тому локальний рівень переважно відповідає за реакцію, edge-рівень — за оперативне оброблення та інтеграцію, а верхні рівні — за координацію, аналіз і планування. Це не жорстке правило, але воно добре відображає фізичну природу розподілу функцій у КФВС.

2.4. Централізовані, децентралізовані та розподілені архітектури

У централізованій архітектурі значна частина рішень формується одним центральним вузлом. Перевагою є простіша координація: центральний контролер має цілісне представлення процесу та безпосередньо визначає поведінку підлеглих компонентів.

Недолік полягає у залежності від центрального вузла. Його відмова може вплинути одразу на значну частину системи. Крім того, зі збільшенням кількості компонентів зростає обчислювальне та комунікаційне навантаження.

Якщо працездатність системи потребує одночасної роботи m незалежних послідовно включених функціональних компонентів, імовірність безвідмовної роботи такого ланцюга можна оцінити як:

$$P_c = \prod_{j=1}^m p_j,$$

де; p_j — імовірність безвідмовної роботи окремого компонента; m — кількість компонентів.

Формула справедлива за припущення статистичної незалежності відмов. Вона показує, що додавання послідовно необхідних елементів без резервування знижує загальну надійність структури.

Для незалежних резервних каналів, коли достатньо працездатності хоча б одного з них:

$$P_{\text{рез}} = 1 - \prod_{j=1}^m (1 - p_j).$$

У децентралізованій системі частина рішень передається локальним вузлам. Наприклад, кожна технологічна комірка може самостійно виконувати цикл операцій, а центральна система лише задає завдання.

Розподілена архітектура йде далі: функція може виконуватися декількома взаємодіючими компонентами, а загальний результат формується через обмін інформацією між ними.

Децентралізація зменшує залежність від одного центрального вузла, але породжує іншу проблему — узгодження локальних рішень. Якщо два автономні модулі одночасно претендують на один ресурс, потрібний механізм арбітражу.

Тому у виробничих КФВС поширений гібридний підхід: локальні вузли самостійно виконують технологічні функції, але координація ресурсів і виробничих завдань залишається на вищому рівні [3, 5].

2.5. Сервісно-орієнтований підхід до побудови виробничих систем

Сервісно-орієнтована архітектура — Service-Oriented Architecture, SOA — передбачає, що функціональні можливості компонента надаються іншим компонентам через формально описані сервіси.

Наприклад, роботизований модуль може надавати не набір окремих сигналів «двигун увімкнути», «вісь перемістити», «захват закрити», а сервіс вищого рівня — «перемістити деталь з позиції А на позицію Б». Внутрішня реалізація операції приховується від зовнішнього споживача.

Такий підхід зменшує залежність системи від конкретного внутрішнього устрою компонента. Якщо робот замінюється іншим, але новий модуль підтримує той самий сервісний інтерфейс, частина програм верхнього рівня може залишитися без змін.

Сервісно-орієнтованість, модульність та інтероперабельність регулярно розглядаються як характерні властивості архітектур кіберфізичного виробництва [3, 6].

Для окремого сервісного запиту повний час відповіді можна умовно подати як:

$$\tau_{\text{серв}} = \tau_{\text{черг}} + \tau_{\text{обр}} + \tau_{\text{мер}},$$

де $\tau_{\text{серв}}$ — час отримання результату сервісного запиту; $\tau_{\text{черг}}$ — очікування оброблення; $\tau_{\text{обр}}$ — власний час виконання сервісу; $\tau_{\text{мер}}$ — затримка мережевого обміну.

З цього випливає обмеження сервісної архітектури: не кожному функцію доцільно оформлювати як віддалений мережевий сервіс. Швидкі контури регулювання зазвичай повинні залишатися всередині локального пристрою, а сервісний інтерфейс використовують для команд вищого рівня.

Сервісний підхід особливо корисний у модульних виробництвах, де обладнання часто змінюється або використовується в різних технологічних маршрутах.

2.6. Модульність, масштабованість та інтероперабельність

Модульна архітектура будується з компонентів, кожний з яких має чітко визначені функції та інтерфейси. Внутрішня реалізація модуля повинна якомога менше впливати на інші частини системи.

Це знижує зв'язаність компонентів. Якщо для заміни одного датчика необхідно змінити програмне забезпечення ПЛК, SCADA, бази даних і MES, архітектура є надмірно залежною від конкретної реалізації.

Масштабованість характеризує здатність архітектури зберігати працездатність при збільшенні кількості обладнання, даних і функцій. При масштабуванні необхідно перевіряти щонайменше мережеве навантаження, обчислювальні ресурси, обсяг збережених даних і кількість одночасних запитів.

При цьому архітектура, яка працює для одного виробничого стенда, не повинна автоматично вважатися придатною для всього підприємства.

Інтероперабельність стосується здатності різнорідних компонентів взаємодіяти без створення спеціального механізму для кожної нової пари пристроїв. Вона має декілька рівнів. На технічному рівні компоненти повинні мати сумісні засоби передавання. На синтаксичному — однаково трактувати формат повідомлень. На семантичному — однаково розуміти зміст параметрів.

Наприклад, передавання числа 1450 через Ethernet означає лише технічну передачу даних. Щоб інший компонент правильно використав це число, він повинен знати, що це частота обертання конкретного двигуна, виражена в об/хв, отримана у певний момент часу.

Проблема семантичної сумісності є однією з причин розвитку стандартизованих інформаційних моделей і цифрових представлень активів [1, 7].

2.7. Архітектурні моделі RAMI 4.0 та Industrial Internet Reference Architecture

Референсна архітектура не є готовою схемою конкретного підприємства. Вона задає систему координат, терміни, принципи та типові зв'язки, за допомогою яких можна впорядкувати проєктування. Огляд референсних архітектур цифрового виробництва показує, що різні моделі мають різний рівень деталізації та різне призначення, тому вибір моделі залежить від задачі [4].

Однією з найвідоміших є Reference Architectural Model Industrie 4.0 — RAMI 4.0.

RAMI 4.0 має тривимірну структуру. Перша вісь відображає рівні ієрархії виробничої системи. Друга пов'язана з життєвим циклом та потоком створення цінності. Третя задає функціональні шари архітектури.

На осі шарів виділяються:

Asset — фізичний актив;

Integration — перетворення фізичної інформації у цифрове представлення;

Communication — комунікація;

Information — інформаційне представлення;

Functional — функціональна логіка;

Business — бізнес-процеси.

Така структура дозволяє розглядати один і той самий виробничий компонент одночасно з точки зору його фізичної реалізації, інформаційного представлення та місця у виробничій ієрархії. RAMI 4.0 не слід трактувати як електричну або мережеву схему. Це координатна модель для систематизації функцій, активів, інформації та стандартів [1, 4, 5].

Інший підхід використовується в Industrial Internet Reference Architecture — IIRA. Вона розглядає промислову систему через чотири взаємопов'язані архітектурні точки зору: Business, Usage, Functional та Implementation [7].

Business viewpoint визначає цілі, зацікавлені сторони та очікуваний результат створення системи.

Usage viewpoint описує, хто і яким чином використовує функціональні можливості системи.

Functional viewpoint задає функціональні компоненти, їх взаємодію та інтерфейси.

Implementation viewpoint пов'язує функціональну структуру з конкретними технологіями, пристроями, мережами та програмними засобами.

RAMI 4.0 та IIRA не суперечать одна одній. Вони акцентують різні аспекти складних промислових систем. Огляд архітектур CPS показує, що проблема полягає не у виборі «єдиної правильної» моделі, а у правильному співвіднесенні референсної архітектури з цілями конкретного проєкту [1, 4].

2.8. Вибір архітектури залежно від структури технологічного процесу

Архітектуру КФВС не можна вибрати без аналізу фізичного об'єкта. Одна й та сама архітектурна схема може бути прийнятною для повільного дискретного виробництва і непридатною для високодинамічного процесу.

Перший критерій — час реакції. Якщо функція повинна виконуватися за мілісекунди, її необхідно розташовувати близько до фізичного об'єкта.

Другий — наслідки відмови. Критичні функції не повинні залежати від одного мережевого вузла або віддаленого сервісу без резервування.

Третій — частота реконфігурації. Якщо виробнича лінія роками випускає один продукт, жорстко задана ієрархія може бути цілком виправданою. Якщо склад обладнання часто змінюється, більшу перевагу мають модульні й сервісно-орієнтовані рішення.

Четвертий — обсяг і характер даних. Передавати всі необроблені вимірювання на сервер не завжди раціонально. Частина даних може агрегуватися або аналізуватися на edge-рівні.

П'ятий — інтеграція з іншими системами. Якщо обладнання має взаємодіяти з MES, логістичними системами та цифровими двійниками, інформаційні інтерфейси потрібно передбачити на початковій стадії проєктування.

Під час порівняння декількох архітектурних варіантів можна використовувати метод зваженої суми критеріїв:

$$Q_j = \sum_{i=1}^M w_i q_{ij},$$

де Q_j — інтегральна оцінка j -го варіанта архітектури; M — кількість критеріїв; w_i — ваговий коефіцієнт критерію; q_{ij} — нормована оцінка варіанта за відповідним критерієм.

Для вагових коефіцієнтів приймають умову:

$$\sum_{i=1}^M w_i = 1.$$

При цьому сам результат багатокритеріального розрахунку не повинен підміняти інженерний аналіз. Якщо варіант має високу інтегральну оцінку, але не забезпечує обов'язкової часової або безпекової вимоги, його потрібно відхилити незалежно від суми балів.

Одним з додаткових критеріїв може бути готовність системи до роботи:

$$A = \frac{T_{\text{прац}}}{T_{\text{прац}} + T_{\text{відн}}},$$

де A — коефіцієнт готовності; $T_{\text{прац}}$ — середня тривалість працездатного стану; $T_{\text{відн}}$ — середня тривалість відновлення після відмови.

Для виробничої КФВС цей показник дозволяє оцінити не тільки надійність окремого пристрою, а наслідки архітектурних рішень. Наприклад, централізація декількох функцій на одному сервері зменшує кількість апаратних вузлів, але підвищує значущість його відмови.

Отже, вибір архітектури є компромісом між швидкодією, надійністю, гнучкістю, масштабованістю, складністю інтеграції та вартістю. Універсальної архітектури, однаково придатної для всіх виробничих процесів, не існує. Референсні моделі задають принципи організації, але конкретна структура повинна формуватися виходячи з фізики технологічного процесу та функціональних вимог.

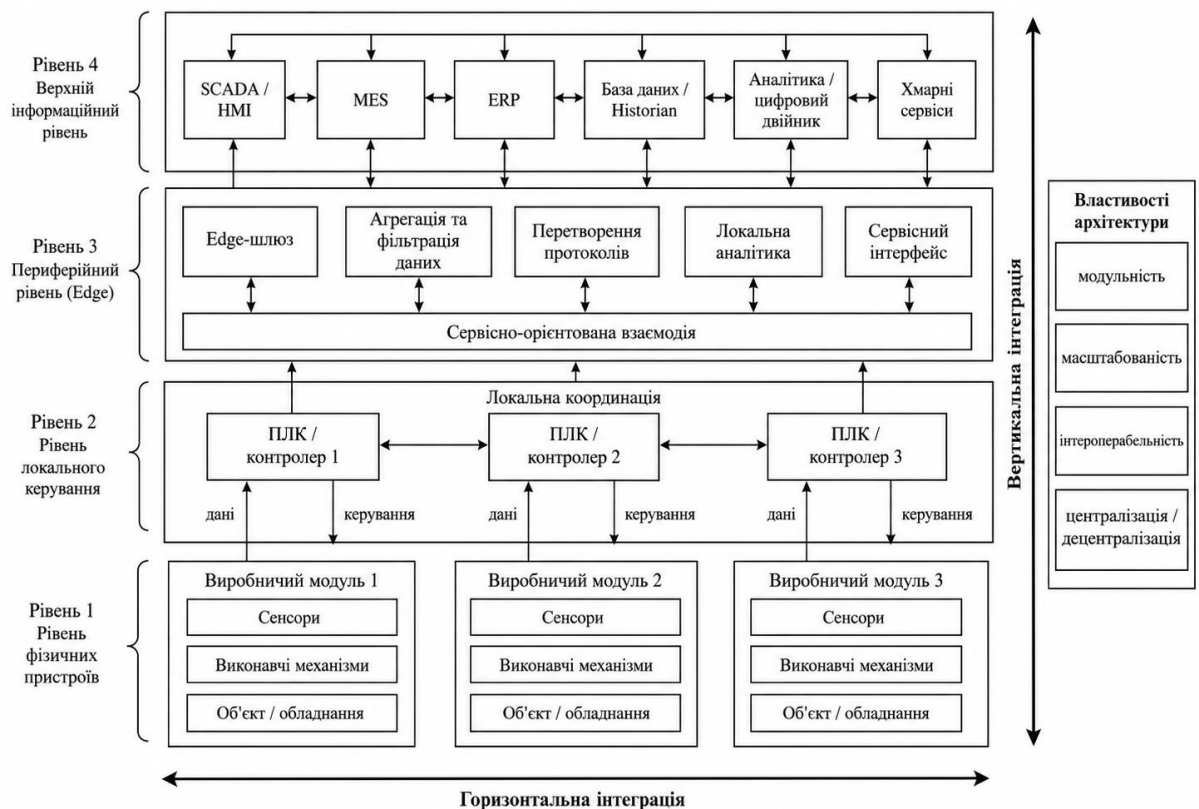


Рисунок – Архітектура кіберфізичної виробничої системи

Рис. 2.1 – Архітектура кіберфізичної виробничої системи

Висновки до лекції 2

Архітектура кіберфізичної виробничої системи визначає склад компонентів, розподіл функцій, інформаційні та керувальні зв'язки, місце виконання обчислень і правила взаємодії окремих підсистем. Її проєктування повинно передувати вибору конкретних програмних і апаратних засобів.

Функціональна декомпозиція дозволяє розділити вимірювання, керування, координацію, оброблення даних, діагностування та виробниче управління. Часово-критичні функції доцільно виконувати локально, тоді як аналітичні та координаційні задачі можуть бути передані на edge-, серверний або корпоративний рівень.

Централізована архітектура спрощує координацію, але підвищує залежність від центрального вузла. Децентралізована і розподілена структури збільшують автономність та гнучкість, однак потребують формалізованих механізмів узгодження локальних рішень.

Модульність і сервісно-орієнтований підхід знижують залежність системи від внутрішньої реалізації окремих компонентів. Їх практична

цінність проявляється лише за наявності чітких інтерфейсів і семантично узгоджених даних.

RAMI 4.0 та IIRA надають референсні засоби структурування складних промислових систем, але не визначають готову архітектуру конкретного виробництва. Остаточне рішення повинно враховувати динаміку фізичного процесу, часові обмеження, вимоги до надійності, характер інформаційних потоків та можливість подальшої реконфігурації.

Контрольні питання

1. Що визначає архітектура кіберфізичної виробничої системи?
2. Для чого застосовується функціональна декомпозиція?
3. Які функції доцільно залишати на локальному рівні?
4. Яке призначення edge-рівня у КФВС?
5. У чому полягають переваги та недоліки централізованої архітектури?
6. Чим децентралізована архітектура відрізняється від розподіленої?
7. У чому полягає сервісно-орієнтований принцип побудови КФВС?
8. Що розуміють під модульністю та масштабованістю?
9. Які рівні інтероперабельності необхідно враховувати?
10. Які три осі формують модель RAMI 4.0?
11. Які функціональні шари містить RAMI 4.0?
12. Які точки зору використовуються в IIRA?
13. Чому референсна архітектура не є готовою схемою системи?
14. Які критерії необхідно враховувати при виборі архітектури КФВС?
15. Чому багатокритеріальна оцінка не може замінити перевірку обов'язкових технічних вимог?

Список використаних джерел

1. Pivoto D. G. S., de Almeida L. F. F., Righi R. R., Rodrigues J. J. P. C., Lugli A. B., Alberti A. M. Cyber-physical systems architectures for industrial internet of things applications in Industry 4.0: A literature review. *Journal of Manufacturing Systems*. 2021. Vol. 58. P. 176–192. DOI: 10.1016/j.jmsy.2020.11.017.
2. Dafflon B., Moalla N., Ouzrout Y. The challenges, approaches, and used techniques of CPS for manufacturing in Industry 4.0: a literature review. *The International Journal of Advanced Manufacturing Technology*. 2021. Vol. 113. P. 2395–2412. DOI: 10.1007/s00170-020-06572-4.
3. Napoleone A., Macchi M., Pozzetti A. A review on the characteristics of cyber-physical systems for the future smart factories. *Journal of*

Manufacturing Systems. 2020. Vol. 54. P. 305–335. DOI: 10.1016/j.jmsy.2020.01.007.

4. Kaiser J., McFarlane D., Hawkrigde G., André P., Leitão P. A review of reference architectures for digital manufacturing: Classification, applicability and open issues. *Computers in Industry*. 2023. Vol. 149. Article 103923. DOI: 10.1016/j.compind.2023.103923.

5. Müller T., Kamm S., Löcklin A., White D., Mellinger M., Jazdi N., Weyrich M. Architecture and knowledge modelling for self-organized reconfiguration management of cyber-physical production systems. *International Journal of Computer Integrated Manufacturing*. 2023. Vol. 36, No. 12. P. 1842–1863. DOI: 10.1080/0951192X.2022.2121425.

6. Traganos K., Grefen P., Vanderfeesten I., Erasmus J., Bouladakis G., Bouklis P. The HORSE framework: A reference architecture for cyber-physical systems in hybrid smart manufacturing. *Journal of Manufacturing Systems*. 2021. Vol. 61. P. 461–494. DOI: 10.1016/j.jmsy.2021.09.003.

7. da Rocha H., Abrishambaf R., Pereira J., Espirito Santo A. Integrating the IEEE 1451 and IEC 61499 Standards with the Industrial Internet Reference Architecture. *Sensors*. 2022. Vol. 22, No. 4. Article 1495. DOI: 10.3390/s22041495.

8. Oks S. J., Jalowski M., Lechner M., Mirschberger S., Merklein M., Vogel-Heuser B., Möslin K. M. Cyber-Physical Systems in the Context of Industry 4.0: A Review, Categorization and Outlook. *Information Systems Frontiers*. 2024. Vol. 26. P. 1731–1772. DOI: 10.1007/s10796-022-10252-x.

ЛЕКЦІЯ 3. МЕТОДИ ФОРМАЛІЗАЦІЇ ТА МОДЕЛЮВАННЯ КІБЕРФІЗИЧНИХ ВИРОБНИЧИХ СИСТЕМ

Мета лекції

Мета лекції полягає у формуванні уявлення про математичні та формальні засоби опису кіберфізичних виробничих систем. Розглядаються безперервні, дискретні та гібридні моделі, моделі у просторі станів, дискретно-подієве представлення, кінцеві автомати та мережі Петрі. Окрема увага приділяється узгодженню моделей фізичної та кібернетичної частин, перевірці адекватності та вибору рівня деталізації.

Моделювання КФВС має специфічну складність: одна система одночасно містить фізичні процеси, які розвиваються неперервно в часі, дискретні алгоритми керування, події, мережеві затримки та логічні стани. Через це одна математична модель рідко однаково добре описує всі її складові. У практичних задачах застосовується сукупність взаємопов'язаних моделей різного типу [1–4].

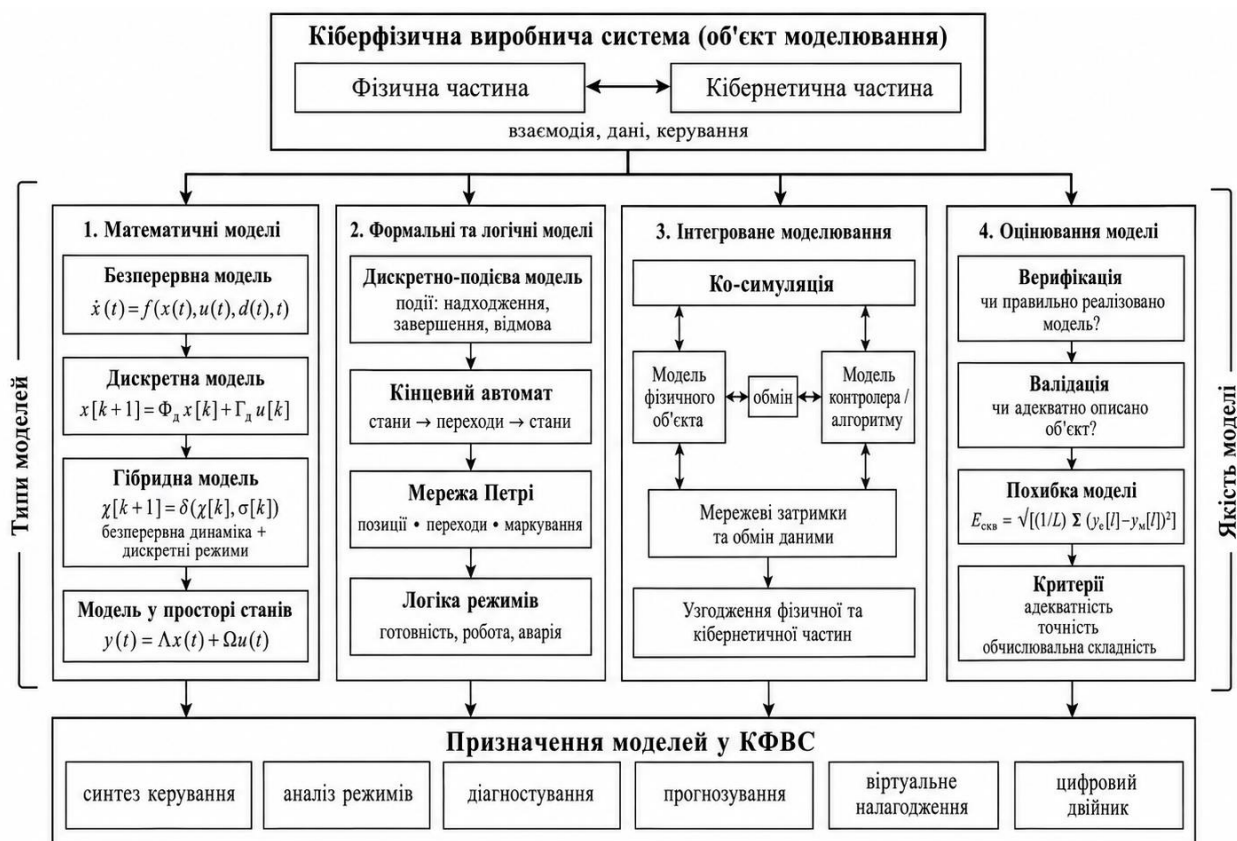


Рис. 3.1 – Методи формалізації та моделювання кіберфізичної виробничої системи

3.1. Модель як основа проєктування кіберфізичної системи

Модель — це формалізоване представлення властивостей реального об'єкта, достатнє для розв'язання конкретної задачі. Вона не повинна відтворювати всі характеристики об'єкта. Надмірна деталізація збільшує обчислювальну складність, але не обов'язково підвищує достовірність результату.

Для одного й того самого виробничого об'єкта можуть використовуватися різні моделі. При розрахунку системи регулювання електропривод описують через механічну та електромагнітну динаміку. Для виробничого планування цей самий привід може бути представлений лише двома станами — «працює» і «не працює». Для діагностування додаються температурні, вібраційні або електричні параметри.

Отже, структура моделі визначається не тільки конструкцією об'єкта, а й метою дослідження.

У КФВС моделі використовуються для:

- синтезу алгоритмів керування;
- аналізу режимів роботи;
- прогнозування стану;
- перевірки логіки виробничих операцій;

- оцінювання продуктивності;
- діагностування;
- віртуального налагодження;
- побудови цифрових двійників.

Модельно-орієнтований підхід дозволяє перевіряти частину проєктних рішень до створення або реконфігурації реального обладнання. У роботах з Model-Based Engineering для CPS модель використовується як спільне представлення структури, поведінки та системи керування, що зменшує кількість неоднозначностей між окремими стадіями розроблення [4, 5].

Проте модель не є самою фізичною системою. Будь-який результат моделювання справедливий лише в межах прийнятих припущень, діапазону параметрів та режимів, для яких модель була сформована і перевірена.

3.2. Моделі безперервних, дискретних та гібридних процесів

За характером зміни стану в часі моделі КФВС можна поділити на безперервні, дискретні та гібридні.

Безперервна модель використовується, якщо основні фізичні координати змінюються неперервно. Такими процесами є нагрівання, рух механічної системи, зміна тиску, швидкості, струму або витрати.

Для лінійного наближення модель стану можна записати:

$$\dot{\mathbf{x}}(t) = \mathbf{\Phi}\mathbf{x}(t) + \mathbf{\Gamma}\mathbf{u}(t),$$

де $\mathbf{\Phi}$ — матриця власної динаміки системи; $\mathbf{\Gamma}$ — матриця впливу керування.

Вихідні величини визначаються як:

$$\mathbf{y}(t) = \mathbf{\Lambda}\mathbf{x}(t) + \mathbf{\Omega}\mathbf{u}(t),$$

де $\mathbf{\Lambda}$ — матриця зв'язку стану з вихідними величинами; $\mathbf{\Omega}$ — матриця безпосереднього впливу керування на вихід.

Лінійна модель придатна не для кожного режиму. Реальні виробничі об'єкти містять насичення, люфти, сухе тертя, обмеження швидкості, нелінійні характеристики та перемикання режимів. Тому лінеаризація використовується в околі певної робочої точки, якщо похибка такого наближення є допустимою.

Дискретна модель описує стан лише в окремі моменти часу. Для лінійного випадку:

$$\mathbf{x}[k+1] = \mathbf{\Phi}_d\mathbf{x}[k] + \mathbf{\Gamma}_d\mathbf{u}[k],$$

де $\mathbf{\Phi}_d$ — матриця переходу дискретної системи; $\mathbf{\Gamma}_d$ — матриця дискретного керування.

Дискретне представлення природне для цифрових контролерів, оскільки вимірювання й обчислення виконуються з певним періодом.

Однак дискретність часу не слід плутати з дискретністю стану. Температура, виміряна кожну секунду, залишається безперервною фізичною величиною, хоча цифрова система отримує її дискретну часову послідовність.

Гібридна модель поєднує неперервну динаміку та дискретні переходи. Такий опис особливо характерний для виробництва. Наприклад, конвеєр має неперервну швидкість, але режими «пуск», «робота», «гальмування» і «аварійна зупинка» є дискретними.

Зміну режиму можна представити:

$$\chi[k+1] = \delta(\chi[k], \sigma[k]),$$

де $\chi[k]$ — дискретний режим системи; δ — функція переходу між режимами; σ — подія, що викликає або дозволяє перехід.

У межах конкретного режиму фізична динаміка може описуватися окремою функцією:

$$\dot{\mathbf{x}}(t) = \mathbf{f}_{\chi}(\mathbf{x}(t), \mathbf{u}(t), \mathbf{d}(t), t),$$

де f_{χ} — модель неперервної динаміки для відповідного дискретного режиму.

Саме гібридний характер є однією з причин складності моделювання КФВС: необхідно одночасно правильно описати фізичну динаміку та умови переключення між режимами. У дослідженнях smart manufacturing гібридне моделювання також застосовується для поєднання моделей, заснованих на фізичних закономірностях, із моделями, отриманими з даних [6].

3.3. Моделі стану виробничих об'єктів

Модель у просторі станів використовується тоді, коли поведінку об'єкта доцільно описувати через мінімальний або достатній набір внутрішніх координат.

Для технологічної установки стан може визначатися температурою, тиском і запасом речовини. Для електропривода — струмом, швидкістю та положенням. Для виробничої комірки поняття стану може бути розширене логічними ознаками: наявністю деталі, готовністю робота, станом захисних пристроїв.

Стан має містити достатню інформацію, щоб за відомими зовнішніми впливами визначити подальшу поведінку моделі.

Не всі координати стану доступні для прямого вимірювання. У такому випадку використовується оцінювання стану за вимірюваними вихідними величинами.

Похибку оцінювання можна записати:

$$\varepsilon(t) = \mathbf{x}(t) - \hat{\mathbf{x}}(t),$$

де $\varepsilon(t)$ — вектор похибки оцінювання стану.

Для КФВС це суттєво, оскільки цифровий алгоритм часто працює не з повним фізичним станом, а з його розрахунковою оцінкою.

При виборі координат стану необхідно уникати зайвого ускладнення. Якщо певна змінна не впливає на результат дослідження в потрібному діапазоні, її включення до моделі може збільшити час обчислення без практичної користі.

З іншого боку, надмірне спрощення здатне приховати критичну властивість. Наприклад, модель електропривода без температурної складової може бути достатньою для аналізу короткого перехідного процесу, але непридатною для прогнозування теплового перевантаження.

3.4. Диференціальні та різницеві моделі

Диференціальні рівняння є основним засобом опису безперервної динаміки фізичних компонентів. Їх структура впливає із законів збереження маси та енергії, рівнянь електричних кіл, механіки, теплопередачі та інших фізичних закономірностей.

Простий процес першого порядку можна подати:

$$T_0 \frac{dy(t)}{dt} + y(t) = K_0 u(t),$$

де T_0 — стала часу об'єкта; K_0 — коефіцієнт передачі об'єкта.

Такою моделлю в певному діапазоні можуть наближено описуватися теплові, гідравлічні або електромеханічні процеси. Її перевага полягає у простоті, але вона не відображає складних нелінійних ефектів.

Для цифрової реалізації безперервні рівняння перетворюються на різницеві. Найпростіше наближення похідної має вигляд:

$$\dot{y}[k] \approx \frac{y[k] - y[k-1]}{T_d},$$

Точність такого наближення залежить від співвідношення між періодом дискретизації та характерним часом зміни процесу. Надто великий період може приховувати швидкі процеси, а надмірне його зменшення підвищує вимоги до обчислень та збільшує обсяг даних.

При моделюванні КФВС різницеві рівняння також дозволяють явно враховувати затримки:

$$y[k] = \mathcal{F}(\mathbf{x}[k], \mathbf{u}[k - \kappa]),$$

де $\mathcal{F}$ — дискретна функція моделі; κ — затримка, виражена кількістю дискретних кроків.

Це корисно для мережевих систем керування, де команда може надходити до виконавчого пристрою не в тому самому циклі, в якому була сформована.

3.5. Дискретно-подієве моделювання

Для багатьох виробничих задач детальна фізична динаміка окремої машини не є визначальною. Якщо потрібно оцінити продуктивність лінії, черги, завантаження обладнання або час проходження замовлення, доцільніше використовувати дискретно-подієві моделі.

У такій моделі стан змінюється при настанні події:

- надходженні деталі;
- завершенні операції;
- звільненні ресурсу;
- виникненні відмови;
- завершенні ремонту;
- надходженні транспортного засобу.

Між подіями модельний час може просуватися без розрахунку кожного проміжного фізичного стану. Через це дискретно-подієва модель часто потребує значно менше обчислень, ніж детальна динамічна модель.

Наприклад, для оцінювання потоку виробів не обов'язково обчислювати миттєвий струм двигуна конвеєра. Достатньо знати тривалість транспортування, місткість ділянки та моменти надходження виробів.

Проте для розподілених КФВС звичайного дискретно-подієвого опису може бути недостатньо, оскільки поведінка залежить не лише від потоку матеріалу, а й від автономних рішень окремих компонентів. У таких випадках використовують агентне або гібридне моделювання. Огляд Рааре, van Eekelen та Reniers показує, що поєднання дискретно-подієвої й агентної симуляції дозволяє краще відображати одночасно фізичний виробничий потік та розподілену логіку керування CPPS [7].

3.6. Кінцеві автомати та мережі Петрі

Кінцевий автомат зручний для формалізації системи, поведінка якої описується обмеженою кількістю станів та правилами переходу між ними.

Його можна представити:

$$\mathcal{M}_A = (\mathcal{S}, \Sigma, \delta, \chi_0, \mathcal{S}_{\text{доп}}),$$

де $\mathcal{M}_A$ — кінцевий автомат; $\mathcal{S}$ — множина можливих станів; Σ — множина вхідних подій; χ_0 — початковий стан; $\mathcal{S}_{\text{доп}}$ — множина допустимих або завершальних станів.

Наприклад, для виробничого модуля можна задати стани «готовність», «завантаження», «обробка», «розвантаження», «аварія». Перехід між ними дозволяється лише за визначених умов.

Перевага автоматної моделі полягає у наочному описі логіки. Водночас при великій кількості незалежних підсистем число комбінацій станів швидко зростає. Це явище називають вибухом простору станів.

Для процесів із паралельністю, синхронізацією та конкуренцією за ресурси більш придатними є мережі Петрі. У виробничих CPS вони застосовуються для формального опису технологічних послідовностей, взаємодії ресурсів та перевірки властивостей поведінки [3, 8].

Мережу Петрі можна представити:

$$\mathcal{N}_\Pi = (\mathcal{P}, \mathcal{T}, \mathcal{F}_\Pi, \mathcal{W}, \mu_0),$$

де $\mathcal{N}_\Pi$ — мережа Петрі; $\mathcal{P}$ — множина позицій; $\mathcal{T}$ — множина переходів; $\mathcal{F}_\Pi$ — множина дуг; $\mathcal{W}$ — функція ваг дуг; μ_0 — початкове маркування.

Позиції зазвичай характеризують умови або стани, переходи — події, а маркери показують поточну ситуацію.

Зміну маркування можна подати:

$$\mu[k+1] = \mu[k] + C\eta[k],$$

де $\mu[k]$ — поточне маркування мережі; C — матриця інцидентності; $\eta[k]$ — вектор спрацювання переходів.

Мережі Петрі дозволяють перевіряти досяжність станів, взаємні блокування, можливість одночасного виконання операцій та використання спільних ресурсів. У роботі Song та співавторів поведінкова модель CPPS побудована на розширеній часовій кольоровій мережі Петрі саме для представлення складної взаємодії компонентів [3].

3.7. Моделювання взаємодії фізичної та кібернетичної частин

Найскладнішою частиною моделі КФВС є не окремий фізичний або програмний компонент, а їх взаємодія.

Фізична модель працює з величинами, що мають фізичний зміст: швидкістю, температурою, тиском, положенням. Кібернетична частина працює з дискретними даними, часовими мітками, повідомленнями та логічними станами.

Між ними необхідно моделювати:

- дискретизацію;
- перетворення одиниць;
- похибки сенсорів;
- затримки;
- втрати даних;
- формування команд;
- обмеження виконавчих механізмів.

Якщо ці зв'язки ігноруються, модель цифрового алгоритму може показувати добрий результат, який не відтвориться на реальному обладнанні.

Frassaroli та Vinco розглядають цю проблему як задачу спільного моделювання гетерогенних фізичних і кібернетичних компонентів CPPS. Їх аналіз показує, що єдине модельне середовище повинно узгоджувати різні часові масштаби й типи динаміки [2].

Для складних систем застосовується ко-симуляція: окремі підсистеми моделюються різними спеціалізованими засобами, але синхронізуються під час спільного виконання. Наприклад, механічна модель може розраховувати рух, а окрема модель контролера — алгоритм керування.

У такому випадку істотною стає похибка синхронізації. Якщо моделі обмінюються даними надто рідко, між ними виникає штучне запізнення. Якщо надто часто — різко зростають обчислювальні витрати.

3.8. Адекватність, точність та обчислювальна складність моделей

Після побудови моделі необхідно перевірити, чи відтворює вона властивості реального об'єкта з точністю, достатньою для поставленої задачі.

Потрібно розрізняти верифікацію та валідацію.

Верифікація відповідає на питання: чи правильно реалізовано саму математичну або алгоритмічну модель?

Валідація відповідає на інше питання: чи достатньо правильно модель описує реальний об'єкт?

Модель може бути безпомилково реалізована програмно, але неадекватно описувати фізичний процес через неправильні припущення.

Для кількісного порівняння експериментальних і модельних даних можна застосовувати середньоквадратичну похибку:

$$E_{\text{СКВ}} = \sqrt{\frac{1}{L} \sum_{\ell=1}^L (y_e[\ell] - y_m[\ell])^2},$$

де $E_{\text{СКВ}}$ — середньоквадратична похибка моделі; L — кількість порівнюваних відліків; ℓ — номер відліку; y_e — експериментальне значення; y_m — значення, отримане за моделлю.

Менше значення похибки свідчить про ближче числове співпадіння, але не гарантує фізичної правильності моделі. Дві різні моделі можуть однаково добре описувати один експеримент і давати різні результати за межами перевіреного режиму.

Тому адекватність потрібно оцінювати в декількох режимах, включаючи ті, які передбачається використовувати під час реальної експлуатації.

Існує також компроміс між точністю та обчислювальною складністю. Детальна модель може бути корисною для офлайн-дослідження, але непридатною для алгоритму, який повинен працювати в реальному часі.

Якщо час розрахунку одного модельного кроку перевищує допустимий період його виконання, така модель не може безпосередньо використовуватися у відповідному контурі:

$$T_{\text{роз}} \leq T_{\text{вик}}$$

де $T_{\text{роз}}$ — фактичний час розрахунку модельного кроку; $T_{\text{вик}}$ — максимально допустимий інтервал його виконання.

Отже, найскладніша модель не обов'язково є найкращою. Для проектування КФВС потрібна модель мінімальної достатньої складності: вона повинна відтворювати властивості, критичні для поставленої задачі, але не містити деталей, які лише збільшують обчислювальні витрати.

Висновки до лекції 3

Моделювання кіберфізичної виробничої системи потребує одночасного опису фізичної динаміки, дискретних алгоритмів, технологічних подій та інформаційної взаємодії. Через це універсального виду моделі для всіх рівнів КФВС не існує.

Безперервні та різницеві моделі придатні для опису динаміки фізичних об'єктів і систем автоматичного керування. Дискретно-подієві моделі доцільні при дослідженні виробничих потоків, черг і використання ресурсів. Кінцеві автомати описують логіку переходів між станами, а мережі Петрі дозволяють формалізувати паралельність, синхронізацію та конфлікти ресурсів.

Для КФВС особливе значення мають гібридні моделі, оскільки реальний фізичний процес є безперервним, а цифрові алгоритми та технологічні режими мають дискретний характер.

Модель має оцінюватися не за максимальною деталізацією, а за відповідністю поставленій задачі. Її адекватність необхідно перевіряти за

експериментальними даними, а обчислювальна складність повинна відповідати часовим обмеженням системи.

Контрольні питання

1. Для чого використовуються моделі при проектуванні КФВС?
2. Чим безперервна модель відрізняється від дискретної?
3. Що розуміють під гібридною моделлю?
4. Яку інформацію містить вектор стану об'єкта?
5. Для яких задач застосовуються диференціальні рівняння?
6. Яку роль відіграє період дискретизації?
7. Коли доцільно використовувати дискретно-подієве моделювання?
8. Що являє собою кінцевий автомат?
9. Які елементи входять до мережі Петрі?
10. Для яких виробничих процесів мережі Петрі мають переваги перед простими автоматами?
11. У чому полягає ко-симуляція?
12. Чим верифікація моделі відрізняється від її валідації?
13. Як може оцінюватися похибка моделі?
14. Чому підвищення деталізації не завжди покращує модель?
15. Яке обмеження накладає режим реального часу на складність моделі?

Список використаних джерел

1. Oks S. J., Jalowski M., Lechner M. et al. Cyber-Physical Systems in the Context of Industry 4.0: A Review, Categorization and Outlook. *Information Systems Frontiers*. 2024. Vol. 26. P. 1731–1772. DOI: 10.1007/s10796-022-10252-x.
2. Fraccaroli E., Vinco S. Modeling Cyber-Physical Production Systems with SystemC-AMS. *IEEE Transactions on Computers*. 2023. Vol. 72, No. 7. P. 2039–2051. DOI: 10.1109/TC.2022.3226567.
3. Song S., Jiang Z., Ma J., Li Q., Wang Q. Modelling and platform application of the behaviour of a cyber physical production system. *International Journal of Computer Integrated Manufacturing*. 2021. Vol. 34, No. 12. P. 1305–1326. DOI: 10.1080/0951192X.2021.1972458.
4. Parant A., Gellot F., Zander D. et al. Model-based engineering for designing cyber-physical systems from product specifications. *Computers in Industry*. 2023. Vol. 145. Article 103808. DOI: 10.1016/j.compind.2022.103808.
5. Campo K. X., Teper T., Eaton C. E., Shipman A. M., Bhatia G., Mesmer B. Model-based systems engineering: Evaluating perceived value, metrics, and evidence through literature. *Systems Engineering*. 2023. Vol. 26. P. 104–129. DOI: 10.1002/sys.21644.

6. Yang S., Navarathna P., Ghosh S., Bequette B. W. Hybrid Modeling in the Era of Smart Manufacturing. *Computers & Chemical Engineering*. 2020. Vol. 140. Article 106874. DOI: 10.1016/j.compchemeng.2020.106874.

7. Paape N., van Eekelen J. A. W. M., Reniers M. A. Review of simulation software for cyber-physical production systems with intelligent distributed production control. *International Journal of Computer Integrated Manufacturing*. 2024. Vol. 37, No. 5. P. 589–611. DOI: 10.1080/0951192X.2023.2228270.

8. Wang G., Li D., Tu Y., Zhang C., Li F., Wang S. A product-process-resource based formal modelling framework for customized manufacturing in cyber-physical production systems. *International Journal of Computer Integrated Manufacturing*. 2022. Vol. 35, No. 6. P. 598–618. DOI: 10.1080/0951192X.2021.1992662.

ЛЕКЦІЯ 4. ІДЕНТИФІКАЦІЯ, СЕНСОРНІ СИСТЕМИ ТА ФОРМУВАННЯ ВИРОБНИЧИХ ДАНИХ

Мета лекції

Мета лекції полягає у формуванні уявлення про принципи отримання, перетворення та первинного оброблення даних у кіберфізичних виробничих системах. Розглядаються структура вимірювального каналу, промислові сенсори, дискретизація та квантування сигналів, похибки вимірювань, методи фільтрації, ідентифікація параметрів виробничих об'єктів і контекстуалізація даних.

Для КФВС вимірювальні дані є основним каналом зв'язку між фізичним об'єктом та його цифровим представленням. Алгоритм керування, система діагностування або цифровий двійник не мають безпосереднього доступу до фізичного стану обладнання. Вони використовують дані, сформовані сенсорами, перетворювачами та програмними засобами. Тому властивості вимірювального каналу безпосередньо обмежують достовірність цифрового представлення виробничого процесу [1–3].

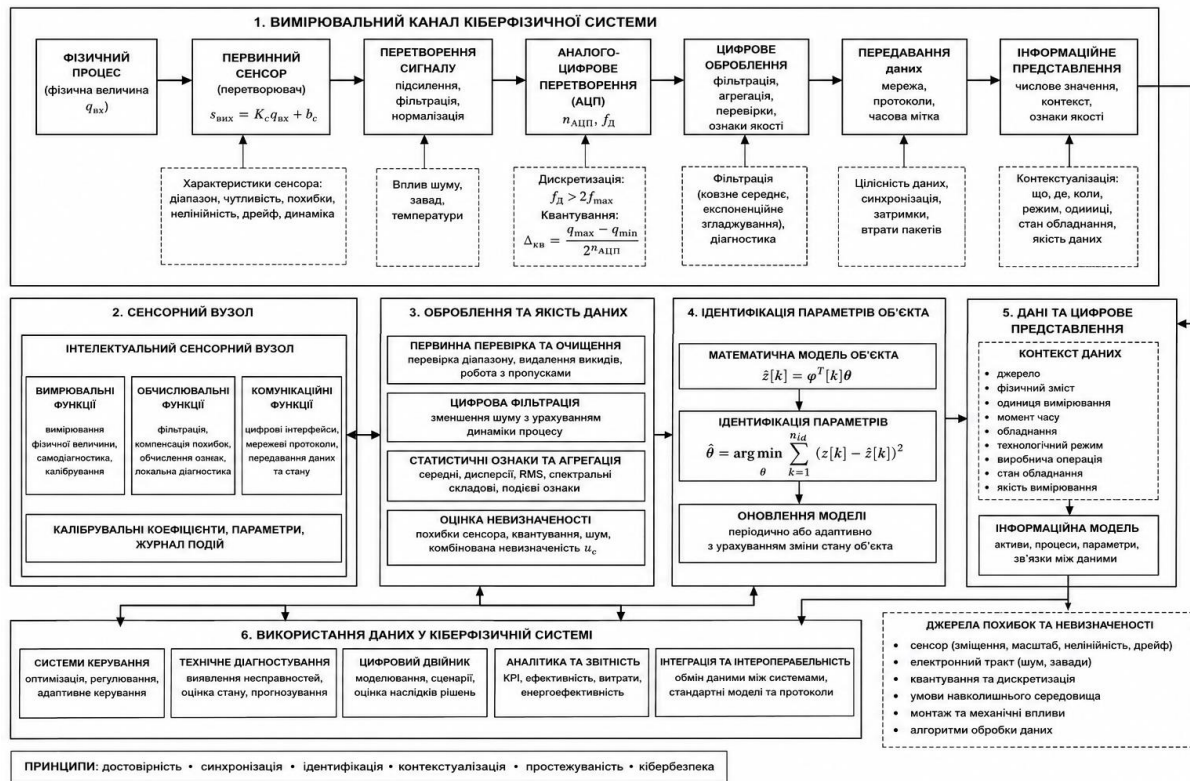


Рисунок 4.1 – Вимірювальні канали кіберфізичних систем

4.1. Структура вимірювального каналу кіберфізичної системи

Вимірювальний канал КФВС являє собою послідовність технічних і програмних перетворень, у результаті яких фізична величина перетворюється на цифрове значення, придатне для використання алгоритмами керування та аналізу.

У загальному випадку можна виділити такі етапи:

фізична величина → первинний сенсор → перетворення сигналу → нормалізація → аналого-цифрове перетворення → цифрове оброблення → передавання → інформаційне представлення.

Наприклад, при вимірюванні температури підшипника первинний перетворювач реагує на температуру, формує електричний сигнал, після чого сигнал підсилюється, фільтрується та перетворюється у цифровий код. Контролер отримує вже не температуру як фізичний процес, а числове представлення результату вимірювання.

Для лінійного сенсора статичну характеристику в робочому діапазоні можна наближено записати:

$$s_{вих} = K_c q_{вх} + b_c,$$

де $s_{вих}$ — вихідний сигнал сенсора; K_c — коефіцієнт перетворення сенсора; $q_{вх}$ — вимірювана фізична величина; b_c — зміщення нульового рівня.

Реальна характеристика може відхилятися від лінійної через нелінійність, температурний дрейф, гістерезис, старіння елементів та вплив зовнішнього середовища. Тому паспортна точність сенсора є лише однією зі складових загальної достовірності вимірювального каналу.

Вимірювальний канал також має динамічні властивості. Навіть якщо сенсор точно відображає сталу величину, він не обов'язково миттєво реагує на її швидку зміну. Це особливо істотно для температурних, газоаналітичних і деяких механічних сенсорів.

Отже, під час вибору сенсора необхідно враховувати не тільки діапазон і статичну похибку, а й швидкодію, смугу пропускання, умови монтажу та характер фізичного процесу.

4.2. Датчики та інтелектуальні сенсорні вузли

У промислових КФВС використовуються сенсори температури, тиску, витрати, рівня, положення, швидкості, сили, моменту, вібрації, струму, напруги, вологості, концентрації речовин та інших параметрів. Вибір сенсора визначається фізичною природою процесу та інформацією, необхідною для конкретної функції системи.

Сенсор у традиційній системі автоматизації може виконувати лише перетворення фізичної величини у стандартний аналоговий сигнал. Інтелектуальний сенсорний вузол додатково містить обчислювальні та комунікаційні засоби. Він може виконувати локальну фільтрацію, компенсацію температурної похибки, самодіагностику, зберігати калібрувальні коефіцієнти та передавати цифрові дані через мережу [1, 4].

Огляди сучасних сенсорних технологій показують, що розвиток промислових сенсорів пов'язаний не тільки з підвищенням точності, а й з інтеграцією локальних обчислень і комунікаційних функцій [1]. Інтелектуальний сенсор стає самостійним інформаційним компонентом КФВС, а не лише джерелом аналогового сигналу.

Для сенсорного вузла доцільно розрізняти три групи функцій:

- вимірювальні;
- обчислювальні;
- комунікаційні.

Вимірювальна частина визначає фізичний параметр. Обчислювальна виконує первинну обробку. Комунікаційна передає результат до ПЛК, edge-вузла або іншого компонента.

Таке розділення суттєве при проєктуванні. Якщо кожне необроблене високочастотне вимірювання передавати до центрального сервера, мережа може отримати значне навантаження. Якщо частину оброблення

виконувати безпосередньо біля сенсора, можна передавати вже ознаки або агреговані показники.

Для задач технічного діагностування це може бути, наприклад, не повна часова реалізація вібраційного сигналу, а середньоквадратичне значення, спектральні складові або ознака перевищення заданого рівня.

Однак надмірне локальне оброблення також має недолік: первинні дані можуть бути втрачені, а повторний аналіз іншими методами стане неможливим. Тому спосіб оброблення слід визначати відповідно до призначення даних.

4.3. Аналогові та цифрові вимірювальні сигнали

Фізичні процеси переважно мають безперервний характер, тоді як цифровий контролер працює з числовими значеннями. Перехід між цими представленнями здійснюється шляхом дискретизації та квантування.

Аналоговий сигнал може набувати будь-якого значення в певному діапазоні. Цифровий сигнал представлений скінченною множиною числових кодів.

Дискретизація визначає, коли виконується вимірювання, а квантування — яким із доступних цифрових рівнів буде представлено його значення.

Кількість можливих кодів $n_{\text{АЦП}}$ -розрядного аналого-цифрового перетворювача визначається:

$$N_{\text{код}} = 2^{n_{\text{АЦП}}},$$

де $N_{\text{код}}$ — кількість цифрових кодів АЦП; $n_{\text{АЦП}}$ — розрядність аналого-цифрового перетворювача.

Крок квантування для рівномірного АЦП можна наближено визначити:

$$\Delta_{\text{кв}} = \frac{q_{\text{max}} - q_{\text{min}}}{2^{n_{\text{АЦП}}}},$$

де q_{max} — верхня межа вимірювального діапазону; q_{min} — нижня межа вимірювального діапазону.

Зі збільшенням розрядності крок квантування зменшується. Проте висока розрядність АЦП не гарантує такої самої фактичної точності. Якщо електричний шум сенсорного каналу перевищує один молодший розряд, частина додаткової цифрової роздільної здатності не несе корисної інформації.

Для ідеального рівномірного квантування максимальне відхилення через округлення становить:

$$|e_{\text{KB}}| \leq \frac{\Delta_{\text{KB}}}{2}.$$

Отже, вибір АЦП повинен бути узгоджений із реальними характеристиками сенсора, рівнем шуму та необхідною точністю всієї системи.

4.4. Дискретизація та квантування сигналів

Частота дискретизації визначає часову деталізацію цифрового представлення фізичного процесу. Надто низька частота може призвести до втрати швидких складових сигналу.

Для гармонічно обмеженого сигналу теорема Найквіста–Шеннона задає умову:

$$f_{\text{д}} > 2 f_{\text{max}},$$

де f_{max} — максимальна частота спектральної складової сигналу, яку необхідно коректно відтворити.

Формальне виконання цієї нерівності ще не означає, що обрана частота оптимальна. У реальній вимірювальній системі використовуються аналогові антиаліасингові фільтри, а частоту дискретизації часто обирають із певним запасом.

Якщо, наприклад, вимірюється повільна температура технологічного середовища, немає сенсу зчитувати датчик десятки тисяч разів за секунду. Це збільшить обсяг даних, але практично не додасть інформації.

Для вібраційного сигналу ситуація протилежна. Якщо діагностична ознака знаходиться у високочастотній області спектра, надто мала частота дискретизації може повністю приховати її.

З цим пов'язана проблема аліасингу: частотні складові, що перевищують половину частоти дискретизації, після оцифрування можуть відображатися як хибні нижчі частоти.

Таким чином, вибір частоти дискретизації визначається не типом сенсора як таким, а спектральними властивостями тієї частини процесу, яку необхідно спостерігати.

Додатково необхідно враховувати обсяг формованих даних. Збільшення частоти дискретизації пропорційно збільшує інформаційний потік, що було показано в Лекції 1.

Тому існує компроміс:

часова деталізація $\leftrightarrow$ обсяг даних $\leftrightarrow$ обчислювальне навантаження.

4.5. Похибки вимірювання та невизначеність даних

Результат вимірювання завжди містить певне відхилення від значення фізичної величини. Причинами можуть бути властивості сенсора, електронного тракту, монтажу, навколишнього середовища та алгоритму перетворення.

До поширених складових належать:

- зміщення нуля;
- похибка масштабу;
- нелінійність;
- гістерезис;
- дрейф;
- температурна залежність;
- випадковий шум;
- похибка квантування.

У КФВС ця проблема має додатковий аспект. Дані після вимірювання використовуються іншими алгоритмами, тому невизначеність поширюється вздовж цифрового ланцюга.

Якщо результат залежить від декількох незалежних вхідних величин, стандартну комбіновану невизначеність у спрощеному випадку можна оцінити:

$$u_c = \sqrt{\sum_{p=1}^{n_b} (c_p u_p)^2},$$

де u_c — комбінована стандартна невизначеність; n_b — кількість складових невизначеності; p — номер складової; c_p — коефіцієнт чутливості до відповідної складової; u_p — її стандартна невизначеність.

Це співвідношення передбачає відсутність суттєвої кореляції між складовими. За наявності корельованих величин модель невизначеності ускладнюється.

Для кіберфізичної системи важливо розрізняти значення даних і довіру до цього значення. Наприклад, температура 85 °С, отримана справним відкаліброваним сенсором, і те саме число, сформоване датчиком з діагностичною помилкою, не повинні однаково використовуватися алгоритмом.

Тому сучасний сенсорний вузол може передавати разом із вимірюванням ознаки якості, стан калібрування, діагностичні коди та часову мітку.

4.6. Фільтрація та первинне оброблення вимірювальної інформації

Первинні сенсорні дані часто містять шум, одиничні викиди, пропуски або високочастотні складові, які не пов'язані з досліджуваним технологічним процесом. Перед використанням у системі керування їх необхідно обробити.

Найпростішим способом згладжування є ковзне середнє:

$$\bar{z}[k] = \frac{1}{n_{\phi}} \sum_{r=0}^{n_{\phi}-1} z[k-r],$$

де $\bar{z}[k]$ — відфільтроване значення; $z[k]$ — поточне виміряне значення; n_{ϕ} — довжина вікна фільтра; r — номер елемента всередині вікна.

Збільшення вікна сильніше згладжує шум, але одночасно уповільнює реакцію на реальні зміни процесу.

Інший поширений варіант — експоненціальне згладжування:

$$\tilde{z}[k] = \alpha_{\phi} z[k] + (1 - \alpha_{\phi}) \tilde{z}[k-1],$$

де $\tilde{z}[k]$ — результат експоненціальної фільтрації; α_{ϕ} — коефіцієнт згладжування в межах від 0 до 1.

При меншому значенні α_{ϕ} фільтрація сильніша, але реакція повільніша.

Вибір фільтра повинен враховувати фізичну природу сигналу. Не можна автоматично згладжувати всі різкі зміни: раптовий стрибок може бути не шумом, а реальною аварійною подією.

Первинне оброблення може також включати:

- перевірку фізично допустимого діапазону;
- видалення очевидно некоректних значень;
- компенсацію зміщення;
- перерахунок у фізичні одиниці;
- синхронізацію часових міток;
- визначення статистичних ознак;
- агрегацію.

Інтелектуальні сенсорні системи та predictive maintenance використовують саме таку послідовну підготовку даних перед діагностичним аналізом [4].

4.7. Ідентифікація параметрів виробничого об'єкта

Не всі параметри математичної моделі можна достатньо точно визначити за конструкторською документацією. Частина характеристик

змінюється в процесі експлуатації через зношування, зміну навантаження, температури або властивостей матеріалу.

Ідентифікація полягає у визначенні структури або параметрів моделі за експериментальними даними.

Для параметричної моделі прогнозоване значення можна записати:

$$\hat{z}[k] = \varphi^T[k]\theta,$$

де $\varphi[k]$ — вектор відомих регресійних змінних; θ — вектор параметрів моделі.

Параметри можуть визначатися шляхом мінімізації суми квадратів похибок:

$$J(\theta) = \sum_{k=1}^{n_{id}} (z[k] - \hat{z}[k])^2,$$

де J — критерій ідентифікації; n_{id} — кількість експериментальних відліків, використаних для оцінювання параметрів.

Оцінка параметрів визначається:

$$\hat{\theta} = \arg \min_{\theta} J(\theta).$$

Результат ідентифікації залежить не тільки від математичного методу, а й від експериментальних даних. Якщо об'єкт весь час працює практично в одному режимі, даних може бути недостатньо для надійного визначення всіх параметрів.

Також необхідно розділяти дані, використані для визначення моделі, і дані для її незалежної перевірки. Інакше можна отримати модель, що добре відтворює конкретний експеримент, але погано працює в інших режимах.

У КФВС ідентифікація може виконуватися періодично або адаптивно. Це дозволяє оновлювати цифрове представлення об'єкта при зміні його характеристик. Такий підхід пов'язаний із концепцією виробництва, керованого даними, де дані фізичного процесу використовуються для уточнення моделей та подальших рішень [5].

4.8. Контекстуалізація виробничих даних

Однією з головних відмінностей між звичайним вимірюванням та інформаційним середовищем КФВС є необхідність контекстуалізації даних.

Число саме по собі не є достатнім описом стану виробничого об'єкта. Для його правильного використання необхідно знати:

- джерело;
- фізичний зміст;
- одиницю вимірювання;
- момент отримання;

- технологічний режим;
- виробничу операцію;
- стан обладнання;
- якість вимірювання.

Наприклад, значення віброшвидкості 4 мм/с може мати різне значення для двох різних машин або для однієї машини при різних швидкостях обертання.

Отже, дані повинні бути пов'язані з цифровим описом активу та технологічного процесу.

Контекстуалізація дозволяє перейти від набору розрізнених сигналів до інформаційної моделі:

«що виміряно?» → «де?» → «коли?» → «для якого обладнання?» → «у якому режимі?»

Це особливо істотно для великих КФВС, де одна база може містити десятки тисяч змінних з різних підсистем.

Без семантичного опису інтеграція систем різних виробників потребує ручного встановлення відповідностей між параметрами. Зі збільшенням масштабу виробництва такий спосіб стає складним у підтримці.

Саме тому архітектури промислових CPS приділяють значну увагу інформаційним моделям та інтероперабельності [2, 3].

Контекст також потрібний для алгоритмів машинного навчання. Якщо дані за різних технологічних режимів механічно об'єднати в одну вибірку, алгоритм може трактувати нормальну зміну режиму як аномалію. Тому перед статистичним аналізом необхідно встановити, за яких фізичних умов сформовано кожний фрагмент даних.

У цьому сенсі якість даних КФВС визначається не тільки точністю числа. Повноцінний запис повинен бути достовірним, синхронізованим, однозначно ідентифікованим та пов'язаним з технологічним контекстом.

Висновки до лекції 4

Сенсорна підсистема є основним засобом перенесення інформації від фізичного виробничого процесу до цифрової частини КФВС. Вона охоплює не тільки первинний датчик, а весь вимірювальний канал: перетворення сигналу, аналого-цифрове перетворення, первинне оброблення, передавання та формування інформаційного представлення.

Вибір частоти дискретизації та розрядності АЦП повинен відповідати спектральним властивостям сигналу і необхідній точності. Надмірне

збільшення цих параметрів створює додатковий потік даних, але не обов'язково підвищує якість інформації.

Вимірювальні дані містять похибки та невизначеність, тому цифровий алгоритм повинен враховувати не тільки саме числове значення, а й достовірність його отримання. Фільтрація дозволяє зменшити вплив шуму, але надмірне згладжування здатне приховувати реальні швидкі зміни процесу.

Ідентифікація дозволяє визначати або уточнювати параметри математичної моделі за експериментальними даними. Її результат залежить від інформативності експерименту та діапазону режимів, у яких отримано вихідні дані.

Для КФВС числові вимірювання повинні доповнюватися контекстом. Зв'язок параметра з конкретним обладнанням, моментом часу, одиницею вимірювання та технологічним режимом формує основу подальшого діагностування, аналітики й побудови цифрових двійників.

Контрольні питання

1. Які етапи входять до вимірювального каналу КФВС?
2. Чим інтелектуальний сенсор відрізняється від традиційного первинного перетворювача?
3. Яка різниця між дискретизацією та квантуванням?
4. Від чого залежить крок квантування АЦП?
5. Яке обмеження накладає теорема Найквіста–Шеннона?
6. Що таке аліасинг?
7. Які основні джерела похибок промислового вимірювального каналу?
8. Чому висока розрядність АЦП не гарантує високої фактичної точності?
9. Для чого застосовується цифрова фільтрація?
10. Як довжина вікна ковзного середнього впливає на сигнал?
11. У чому полягає задача ідентифікації?
12. Чому дані для ідентифікації та перевірки моделі доцільно розділяти?
13. Що розуміють під контекстуалізацією виробничих даних?
14. Чому числового значення параметра недостатньо для його використання у КФВС?
15. Як якість сенсорних даних впливає на цифрову модель виробничого об'єкта?

Список використаних джерел

1. Javaid M., Haleem A., Singh R. P., Rab S., Suman R. Significance of sensors for Industry 4.0: Roles, capabilities, and applications. *Sensors International*. 2021. Vol. 2. Article 100110. DOI: 10.1016/j.sintl.2021.100110.
2. Pivoto D. G. S., de Almeida L. F. F., Righi R. R., Rodrigues J. J. P. C., Lugli A. B., Alberti A. M. Cyber-physical systems architectures for industrial internet of things applications in Industry 4.0: A literature review. *Journal of Manufacturing Systems*. 2021. Vol. 58. P. 176–192. DOI: 10.1016/j.jmsy.2020.11.017.
3. Dafflon B., Moalla N., Ouzrout Y. The challenges, approaches, and used techniques of CPS for manufacturing in Industry 4.0: a literature review. *The International Journal of Advanced Manufacturing Technology*. 2021. Vol. 113. P. 2395–2412. DOI: 10.1007/s00170-020-06572-4.
4. Pech M., Vrchota J., Bednář J. Predictive Maintenance and Intelligent Sensors in Smart Factory: Review. *Sensors*. 2021. Vol. 21, No. 4. Article 1470. DOI: 10.3390/s21041470.
5. Suvarna M., Yap K. S., Yang W., Li J., Ng Y. T., Wang X. Cyber-Physical Production Systems for Data-Driven, Decentralized, and Secure Manufacturing—A Perspective. *Engineering*. 2021. Vol. 7, No. 9. P. 1212–1223. DOI: 10.1016/j.eng.2021.04.021.
6. Hoffmann M. W., Malakuti S., Grüner S., Finster S., Gebhardt J., Tan R., Schindler T., Gamer T. Developing Industrial CPS: A Multi-Disciplinary Challenge. *Sensors*. 2021. Vol. 21, No. 6. Article 1991. DOI: 10.3390/s21061991.
7. Dijmărescu I. Sustainable, Smart, and Sensing Technologies for Cyber-Physical Manufacturing Systems: A Systematic Literature Review. *Sustainability*. 2021. Vol. 13, No. 10. Article 5495. DOI: 10.3390/su13105495.
8. Tan S. F., Samsudin A. Recent Technologies, Security Countermeasure and Ongoing Challenges of Industrial Internet of Things (IIoT): A Survey. *Sensors*. 2021. Vol. 21, No. 19. Article 6647. DOI: 10.3390/s21196647.

ЛЕКЦІЯ 5. ВИКОНАВЧІ ПРИСТРОЇ ТА ЛОКАЛЬНІ СИСТЕМИ АВТОМАТИЧНОГО КЕРУВАННЯ

Мета лекції

Мета лекції полягає у формуванні уявлення про місце виконавчих пристроїв у кіберфізичній виробничій системі, принципи побудови локальних контурів автоматичного керування та особливості їх взаємодії з

цифровою частиною КФВС. Розглядаються електричні, пневматичні та гідравлічні виконавчі механізми, регульований електропривод, замкнені контури керування, ПІ- та ПІД-регулятори, дискретне керування технологічними операціями, координація декількох локальних контурів і обмеження реальних виконавчих пристроїв.

Виконавчий пристрій є завершальною ланкою інформаційно-керувального циклу. Саме через нього результат цифрового обчислення перетворюється на фізичну дію: переміщення, обертання, зміну тиску, витрати, температури або іншого технологічного параметра. Тому характеристики виконавчої підсистеми безпосередньо визначають фактичну швидкодію та точність КФВС. Навіть досконалий алгоритм не може забезпечити результат, який фізично недоступний конкретному механізму [1, 2].

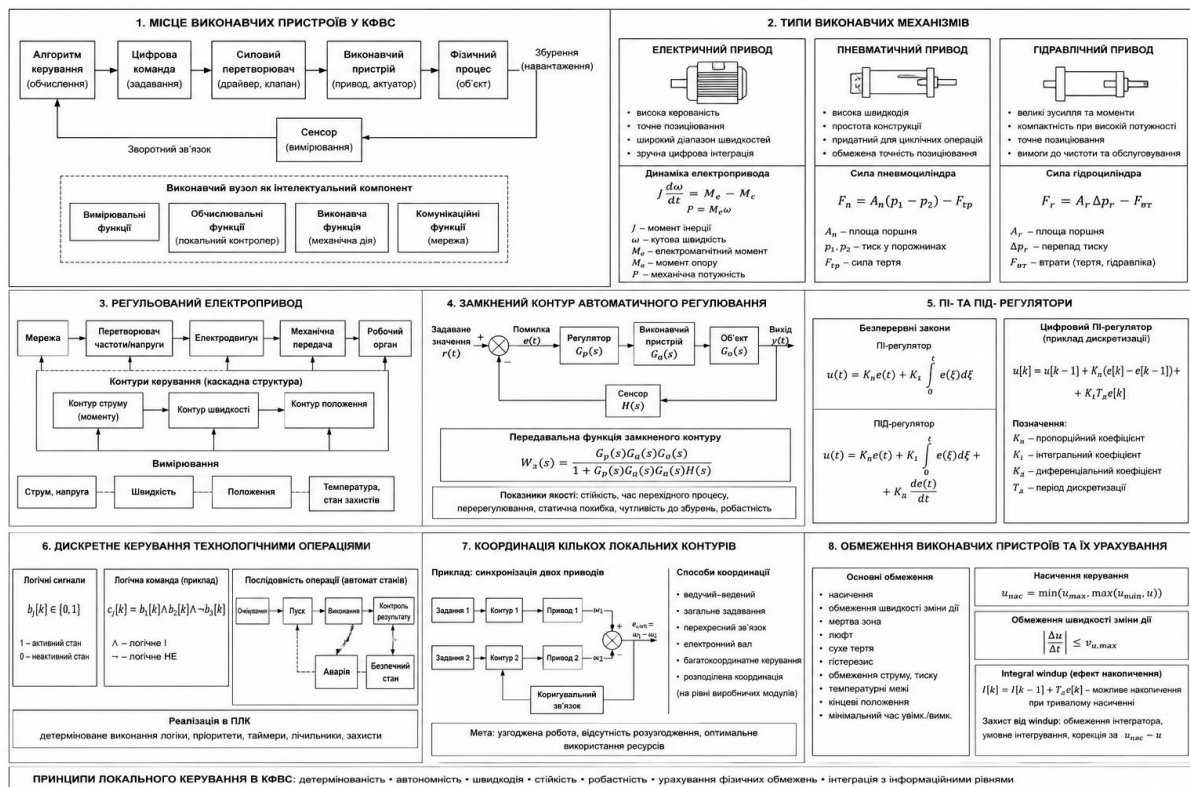


Рисунок 5.1 – Місце виконавчих пристроїв та локальних систем автоматичного керування в КФС

5.1. Місце виконавчих пристроїв у структурі кіберфізичної системи

У структурі КФВС сенсор і виконавчий механізм виконують протилежні за напрямком функції. Сенсор перетворює фізичний стан у цифрову інформацію, тоді як виконавчий пристрій перетворює керувальну команду у фізичний вплив.

Загальну послідовність можна представити так:

алгоритм керування → цифрова команда → силовий перетворювач → виконавчий механізм → фізичний процес.

Між числовим значенням керувальної команди та фактичною зміною об'єкта існує власна динаміка виконавчої системи. У найпростішому наближенні її можна описати аперіодичною ланкою першого порядку:

$$T_a \frac{dy_a(t)}{dt} + y_a(t) = K_a u(t),$$

де T_a — стала часу виконавчого пристрою; $y_a(t)$ — його вихідна фізична величина; K_a — коефіцієнт перетворення.

Це співвідношення показує, що зміна керувального сигналу не призводить до миттєвої зміни фізичного стану. Реальний привід має інерційність, обмежену швидкість і кінцеву потужність.

Виконавчий механізм повинен розглядатися разом із силовою частиною. Для електродвигуна такою частиною є перетворювач напруги або частоти, для пневмоциліндра — розподільник і система підготовки повітря, для гідроциліндра — насос, клапани та гідравлічна магістраль.

У кіберфізичній системі виконавчий вузол часто має власні сенсори та локальний контролер. Наприклад, сучасний електропривод може вимірювати струм, напругу, швидкість, температуру силового модуля, стан захистів і передавати ці дані ПЛК. Такий пристрій виконує одночасно фізичну, вимірювальну, обчислювальну і комунікаційну функції.

Саме така інтеграція сенсорів, контролерів і виконавчих елементів є характерною для промислових кіберфізичних систем [1, 2].

5.2. Електричні, пневматичні та гідравлічні виконавчі механізми

Тип виконавчого механізму визначається фізичною природою необхідної дії, діапазоном швидкості, потрібним зусиллям, точністю позиціонування та умовами експлуатації.

Електричні виконавчі механізми

Електричні приводи широко використовуються у виробничих системах для обертального та поступального руху. До них належать асинхронні, синхронні, вентильні та серводвигуни, двигуни постійного струму, крокові двигуни й електромеханічні актуатори.

Основними перевагами електропривода є простота передачі енергії, висока керованість, можливість точного регулювання швидкості та положення, а також зручна інтеграція з цифровими контролерами.

У загальному випадку механічну динаміку електропривода можна подати:

$$J \frac{d\omega(t)}{dt} = M_e(t) - M_c(t),$$

де J — приведений момент інерції; $\omega(t)$ — кутова швидкість; $M_e(t)$ — електромагнітний момент двигуна; $M_c(t)$ — момент опору механізму.

Механічна потужність на валу становить:

$$P = M_e \omega.$$

З рівняння руху видно, що швидкість визначається не лише командою контролера, а різницею рушійного моменту та навантаження. Різка зміна механічного навантаження тому є збуренням для системи регулювання.

Електроприводи мають особливе значення для КФВС, оскільки вони одночасно є виконавчими та інформаційними вузлами. Дані про струми, швидкість, температуру і режим роботи можуть використовуватися не лише для керування, а й для діагностування технічного стану. Сучасні дослідження приводних систем активно використовують саме експлуатаційні дані для виявлення несправностей [7].

Пневматичні виконавчі механізми

Пневматичні приводи використовують енергію стисненого повітря. Їх застосовують у затискних пристроях, сортувальних механізмах, пакувальному обладнанні, захватах, клапанах і транспортних системах.

Силу пневматичного циліндра у спрощеному вигляді можна оцінити:

$$F_n = A_n (p_1 - p_2) - F_{тр},$$

де F_n — результуюча сила пневмоциліндра; A_n — ефективна площа поршня; p_1 і p_2 — тиски у відповідних порожнинах; $F_{тр}$ — сила тертя.

Перевагами пневматичного привода є простота конструкції, висока швидкість переміщення та придатність для великої кількості циклічних операцій. Недоліки пов'язані зі стисливістю повітря, що ускладнює точне позиційне керування, а також з втратами при підготовці та транспортуванні стисненого повітря.

Гідравлічні виконавчі механізми

Гідропривод використовують там, де необхідно створити значні сили або моменти при відносно компактних розмірах виконавчого механізму.

Для гідроциліндра:

$$F_r = A_r \Delta p_r - F_{вт},$$

де F_r — корисна сила гідроциліндра; A_r — робоча площа поршня; Δp_r — перепад тиску; $F_{вт}$ — сила, еквівалентна механічним і гідравлічним втратам.

Гідравлічні приводи застосовуються у пресах, підйомних механізмах, металургійному обладнанні та інших системах з великими навантаженнями.

Вибір між електричним, пневматичним і гідравлічним приводом не можна виконувати лише за величиною створюваного зусилля. Потрібно також враховувати швидкодію, точність, енергоефективність, умови обслуговування, вимоги до чистоти технологічного середовища та можливість цифрового діагностування.

5.3. Регульований електропривод як компонент виробничої системи

Регульований електропривод є одним із найпоширеніших виконавчих компонентів автоматизованого виробництва. На відміну від нерегульованого привода, він дозволяє змінювати швидкість, момент або положення робочого органа відповідно до команди системи керування.

Типова структура включає:

мережа живлення → силовий перетворювач → електродвигун → механічна передача → робочий орган.

Контур керування доповнюється датчиками струму, швидкості або положення.

Для електромеханічної системи важливе співвідношення між електромагнітним моментом і струмом. У спрощеному вигляді:

$$M_e = K_m i_a,$$

де K_m — коефіцієнт моменту; i_a — струм, що формує момент.

У реальних приводах залежність визначається типом електричної машини і способом керування. Для асинхронних та синхронних машин використовуються складніші моделі, зокрема векторне керування або пряме керування моментом.

Локальний приводний контролер зазвичай має каскадну структуру. Внутрішній контур регулює електричні координати, наступний — швидкість, зовнішній — положення. Внутрішні контури мають більшу швидкодію, ніж зовнішні.

Такий принцип дозволяє розділити динаміку системи за часовими масштабами. Контур струму швидко компенсує електричні зміни, регулятор швидкості реагує на механічне навантаження, а позиційний рівень формує потрібну траєкторію руху.

У КФВС привод перестає бути «чорним ящиком», що отримує лише команду пуску. Параметри його фактичної роботи можуть передаватися до

системи верхнього рівня й використовуватися для прогнозування, технічного обслуговування та оцінювання виробничого стану [7].

5.4. Замкнені контури автоматичного регулювання

Основою локального автоматичного керування є принцип зворотного зв'язку. Виміряне значення порівнюється із заданим, а похибка використовується для формування керувальної дії.

Структурно контур містить регулятор, виконавчий пристрій, фізичний об'єкт і сенсор.

Для лінійної системи передавальну функцію замкненого контуру можна записати:

$$W_3(s) = \frac{G_p(s)G_o(s)}{1 + G_p(s)G_o(s)},$$

де $W_3(s)$ — передавальна функція замкненої системи; $G_p(s)$ — передавальна функція регулятора; $G_o(s)$ — передавальна функція об'єкта разом із виконавчою частиною; s — комплексна змінна перетворення Лапласа.

Зворотний зв'язок дозволяє зменшувати вплив збурень і параметричної невизначеності. Разом з тим надмірне підсилення або значні затримки можуть погіршити стійкість.

Тому якість локального контуру характеризується не одним показником. Зазвичай аналізуються:

- стійкість;
- час перехідного процесу;
- перерегулювання;
- статична похибка;
- чутливість до збурень;
- робастність до зміни параметрів.

Теоретичні дослідження зворотного зв'язку показують, що його цінність полягає саме у здатності зменшувати наслідки невизначеності, але ця властивість досягається лише за правильно сформованого контуру [3].

У КФВС локальний замкнений контур доцільно залишати максимально незалежним від верхніх інформаційних рівнів. SCADA, MES або аналітичний сервер можуть змінювати завдання, але швидке регулювання не повинно залежати від постійної доступності цих систем.

5.5. ПІ-, ПІД- та цифрові регулятори

ПІ- та ПІД-регулятори залишаються широко застосовуваними засобами промислового автоматичного регулювання. Причина полягає у

відносно простій структурі, зрозумілому фізичному змісті параметрів і можливості застосування до широкого класу технологічних об'єктів [3–5]. Огляд сучасних методів налаштування показує, що простота структури ПД-регулятора не означає простоти вибору його параметрів: налаштування завжди пов'язане з компромісом між швидкодією, стійкістю та чутливістю до шуму.

Закон П-регулювання має вигляд:

$$u(t) = K_p e(t) + K_i \int_0^t e(\xi) d\xi,$$

де K_i — інтегральний коефіцієнт; ξ — допоміжна змінна інтегрування.

Інтегральна складова накопичує похибку і дозволяє усунути сталу статичну похибку. Однак надто інтенсивна інтегральна дія може збільшувати перерегулювання та тривалість відновлення після насичення виконавчого пристрою.

ПД-закон додає диференціальну складову:

$$u(t) = K_p e(t) + K_i \int_0^t e(\xi) d\xi + K_d \frac{de(t)}{dt},$$

де K_d — диференціальний коефіцієнт.

Диференціальна складова реагує на швидкість зміни похибки і може покращувати демпфування. Одночасно вона чутлива до високочастотного шуму, тому в практичних регуляторах диференціювання зазвичай фільтрується.

Цифровий ПД-регулятор реалізується у дискретному часі. Один із можливих варіантів дискретного П-закону:

$$u[k] = u[k-1] + K_p (e[k] - e[k-1]) + K_i T_d e[k].$$

Такий запис зручний для програмної реалізації, оскільки керувальний сигнал оновлюється на кожному циклі контролера.

Параметри регулятора не слід налаштовувати ізольовано від виконавчого пристрою. Якщо привод має насичення, мертву зону або значну затримку, модель і параметри контуру повинні це враховувати. Огляди PID-керування наголошують, що якість налаштування суттєво залежить від характеристик реального об'єкта і вибраного критерію якості [4, 5].

5.6. Дискретне керування технологічними операціями

Не всі функції виробничої системи є регулюванням неперервних параметрів. Значна частина технологічних операцій має дискретний характер:

- увімкнути двигун;

- відкрити клапан;
- затиснути деталь;
- перемістити механізм;
- чекати сигналу датчика;
- завершити операцію;
- перейти до наступного стану.

Для таких задач використовується логічне та послідовне керування.

Стан дискретного пристрою можна представити:

$$b_j[k] \in \{0,1\}.$$

Наприклад, одиниця може означати «датчик спрацював», а нуль — «датчик не спрацював». Фізичний зміст повинен бути визначений у специфікації сигналу, оскільки для деяких захисних схем логіка може бути інверсною.

Команда може формуватися логічною функцією декількох умов:

$$c_j[k] = b_1[k] \wedge b_2[k] \wedge \neg b_3[k],$$

де $c_j[k]$ — логічна команда; $\wedge$ — операція логічного «І»; $\neg$ — логічне заперечення.

Такі залежності реалізуються у ПЛК, які залишаються основною платформою промислового локального керування. Сучасні дослідження PLC-систем показують, що вимоги Industry 4.0 збільшують складність програмної логіки, але при цьому не повинні знижувати детермінованість і надійність традиційного промислового керування [6].

Послідовна операція може бути описана станами:

очікування → пуск → виконання → контроль результату → завершення

або, при порушенні умов,

виконання → аварія → безпечний стан.

Для КФВС істотно, що технологічна логіка повинна бути відокремлена від інформаційної аналітики. Втрата зв'язку із сервером статистики не повинна блокувати безпечне завершення локальної виробничої операції.

5.7. Координація кількох локальних контурів

Складна виробнича машина рідко має лише один виконавчий механізм. Робот, верстат, транспортна установка або пакувальна машина містять декілька приводів, які повинні працювати узгоджено.

Найпростіший випадок — синхронізація двох координат:

$$e_{\text{син}}(t) = \omega_1(t) - \omega_2(t),$$

де $e_{\text{син}}(t)$ — похибка синхронізації; $\omega_1(t)$ і $\omega_2(t)$ — швидкості першого та другого механізмів.

Якщо кожний привод підтримує власну швидкість незалежно, невеликі відмінності характеристик або навантажень можуть поступово викликати розузгодження.

У координованій системі локальні контури залишаються замкненими, але отримують взаємопов'язані завдання або додаткові коригувальні сигнали.

Розрізняють декілька способів координації:

- ведучий–ведений;
- загальне задавання;
- перехресний зв'язок;
- електронний вал;
- багатокоординатне керування;
- розподілену координацію.

У схемі «ведучий–ведений» одна координата задає режим іншій. Перевагою є простота. Недолік полягає в тому, що похибка ведучого контуру передається залежному.

При перехресному зв'язку використовується безпосередня похибка між координатами, тому система може компенсувати їх взаємне відхилення.

У КФВС координація може виходити за межі одного механізму. Декілька автономних виробничих модулів можуть узгоджувати швидкість роботи відповідно до стану буферів, доступності робота або виробничого завдання. Це вже поєднує локальне автоматичне регулювання з розподіленим виробничим керуванням. Розподілені CPS саме тому потребують узгодження локальних рішень і фізичних виконавчих дій [2, 8].

5.8. Обмеження виконавчих пристроїв та їх урахування в алгоритмах керування

Лінійні математичні моделі часто припускають, що керувальна дія може набувати будь-якого необхідного значення. Реальний виконавчий пристрій завжди має обмеження.

Найпоширенішим є **насичення**. Наприклад, перетворювач частоти не може сформувати необмежену напругу, двигун має максимальний момент, а клапан не може відкритися більш ніж на 100 %.

Обмежений керувальний сигнал можна представити:

$$u_{\text{нас}} = \min(u_{\text{max}}, \max(u_{\text{min}}, u)),$$

де $u_{\text{нас}}$ — фактичний сигнал після насичення; u_{max} — максимально допустиме значення; u_{min} — мінімально допустиме значення.

Якщо ПІ- або ПІД-регулятор тривалий час вимагає значення, що перевищує фізичну межу, інтегральна складова продовжує накопичувати похибку. Після виходу із насичення це може спричинити значне перерегулювання. Такий ефект називають *integral windup*.

Для його зменшення використовуються *anti-windup* механізми: обмеження інтегральної складової, умовне інтегрування або корекція інтегратора за різницею між розрахованим і фактичним керувальним сигналом.

Іншим обмеженням є максимальна швидкість зміни дії:

$$\left| \frac{\Delta u}{\Delta t} \right| \leq v_{u, \text{max}}.$$

Наприклад, механічний клапан не може перейти з повністю закритого стану у повністю відкритий за нульовий час. Аналогічно електропривод має обмеження прискорення та моменту.

Додатково зустрічаються:

- мертва зона;
- люфт;
- сухе тертя;
- гістерезис;
- обмеження температури;
- обмеження струму;
- мінімальний час увімкнення або вимкнення;
- кінцеві механічні положення.

Ігнорування цих факторів може давати суттєву різницю між моделюванням і реальною роботою.

У КФВС обмеження доцільно представляти не тільки в локальному регуляторі, а й на верхньому інформаційному рівні. Наприклад, система виробничої оптимізації не повинна задавати приводу режим, який суперечить його допустимій температурі або поточному діагностичному стану.

Тому виконавчий пристрій повинен передавати не лише фактичний вихідний параметр, а й інформацію про доступність, обмеження, перевантаження та захисти. Такий підхід дозволяє цифровій частині КФВС приймати рішення, фізично реалізовані конкретним обладнанням.

Висновки до лекції 5

Виконавчі пристрої забезпечують перетворення цифрових керувальних команд у фізичну дію і тому є необхідною частиною замкненого циклу КФВС. Їх динаміка, потужність, швидкодія та фізичні обмеження безпосередньо визначають можливості всієї системи керування.

Електричні, пневматичні та гідравлічні виконавчі механізми мають різні області раціонального застосування. Електропривод забезпечує зручне цифрове регулювання та діагностування, пневматичний привод придатний для швидких циклічних операцій, а гідравлічний — для створення значних сил і моментів.

Локальні замкнені контури залишаються основою фізичного керування навіть у складній кіберфізичній системі. Функції, критичні до часу реакції, доцільно виконувати безпосередньо в ПЛК, приводі або іншому локальному контролері.

ПІ- та ПІД-регулятори мають відносно просту математичну структуру, однак їх налаштування повинно враховувати динаміку об'єкта, властивості сенсорного каналу, затримки та обмеження виконавчого механізму.

У багатопровідних системах локальні контури необхідно координувати. Самостійна стабілізація кожної координати не гарантує узгодженої поведінки всієї машини або виробничого модуля.

Фізичні обмеження — насичення, обмеження швидкості, люфт, мертва зона, температурні й силові межі — мають враховуватися на стадії проектування алгоритму. Кіберфізична система повинна оперувати не абстрактною командою, а командою, яка може бути реально виконана конкретним приводом.

Контрольні питання

1. Яку функцію виконує виконавчий пристрій у КФВС?
2. Чому виконавчий механізм не можна вважати безінерційним?
3. Які основні типи виконавчих приводів використовуються у виробничих системах?
4. Які переваги має регульований електропривод?
5. Чим визначається прискорення електромеханічної системи?
6. Яке призначення зворотного зв'язку?
7. Які показники характеризують якість замкненого контуру?
8. Чим ПІ-регулятор відрізняється від ПІД-регулятора?
9. Яке призначення інтегральної складової?
10. Чому диференціальна складова чутлива до шуму?
11. Для яких задач застосовується дискретне логічне керування?

12. Навіщо координувати кілька локальних приводних контурів?
13. Що розуміють під насиченням виконавчого механізму?
14. У чому полягає ефект integral windup?
15. Які обмеження реального привода необхідно враховувати в алгоритмі керування?

Список використаних джерел

1. Napoleone A., Macchi M., Pozzetti A. A review on the characteristics of cyber-physical systems for the future smart factories. *Journal of Manufacturing Systems*. 2020. Vol. 54. P. 305–335. DOI: 10.1016/j.jmsy.2020.01.007.
2. Dafflon B., Moalla N., Ouzrout Y. The challenges, approaches, and used techniques of CPS for manufacturing in Industry 4.0: a literature review. *The International Journal of Advanced Manufacturing Technology*. 2021. Vol. 113. P. 2395–2412. DOI: 10.1007/s00170-020-06572-4.
3. Guo L. Feedback and uncertainty: Some basic problems and results. *Annual Reviews in Control*. 2020. Vol. 49. P. 27–36. DOI: 10.1016/j.arcontrol.2020.04.001.
4. Borase R. P., Maghade D. K., Sondkar S. Y., Pawar S. N. A review of PID control, tuning methods and applications. *International Journal of Dynamics and Control*. 2021. Vol. 9, No. 2. P. 818–827. DOI: 10.1007/s40435-020-00665-4.
5. Somefun O. A., Akingbade K., Dahunsi F. The dilemma of PID tuning. *Annual Reviews in Control*. 2021. Vol. 52. P. 65–74. DOI: 10.1016/j.arcontrol.2021.05.002.
6. Witte M., Sehr M. A., Ugalde I., Neidig J., Niknami M., Hoeme S., Lee E. A. Opportunities for Industrial Control. *IFAC-PapersOnLine*. 2020. Vol. 53, No. 2. P. 7839–7844. DOI: 10.1016/j.ifacol.2020.12.1951.
7. Gonzalez-Jimenez D., del-Olmo J., Poza J., Garramiola F., Madina P. Data-Driven Fault Diagnosis for Electric Drives: A Review. *Sensors*. 2021. Vol. 21, No. 12. Article 4024. DOI: 10.3390/s21124024.
8. Chinnathai M. K., Alkan B. A Digital Life-Cycle Management Framework for Sustainable Smart Manufacturing in Energy Intensive Industries. *Journal of Cleaner Production*. 2023. Vol. 419. Article 138259. DOI: 10.1016/j.jclepro.2023.138259.

ЛЕКЦІЯ 6. ПРОГРАМОВАНІ ЛОГІЧНІ КОНТРОЛЕРИ ТА СИСТЕМИ КЕРУВАННЯ РЕАЛЬНОГО ЧАСУ

Мета лекції

Мета лекції полягає у формуванні уявлення про принципи побудови та функціонування програмованих логічних контролерів у складі кіберфізичних виробничих систем. Розглядаються апаратна і програмна архітектура ПЛК, циклічний принцип виконання програми, сучасні

положення ІЕС 61131-3, організація логічного, послідовного та регульовального керування, поняття реального часу, детермінованість, часові обмеження та особливості розподіленого керування.

У КФВС програмований логічний контролер займає проміжне положення між фізичним процесом і розподіленим інформаційним середовищем. З одного боку, він повинен гарантовано обробляти сигнали датчиків і формувати керувальні дії в задані моменти часу. З іншого — сучасний ПЛК обмінюється даними з приводами, віддаленими модулями введення-виведення, SCADA, edge-вузлами та іншими контролерами. Розширення комунікаційних можливостей не повинно порушувати головну властивість контролера — передбачуване виконання функцій керування [2, 3].

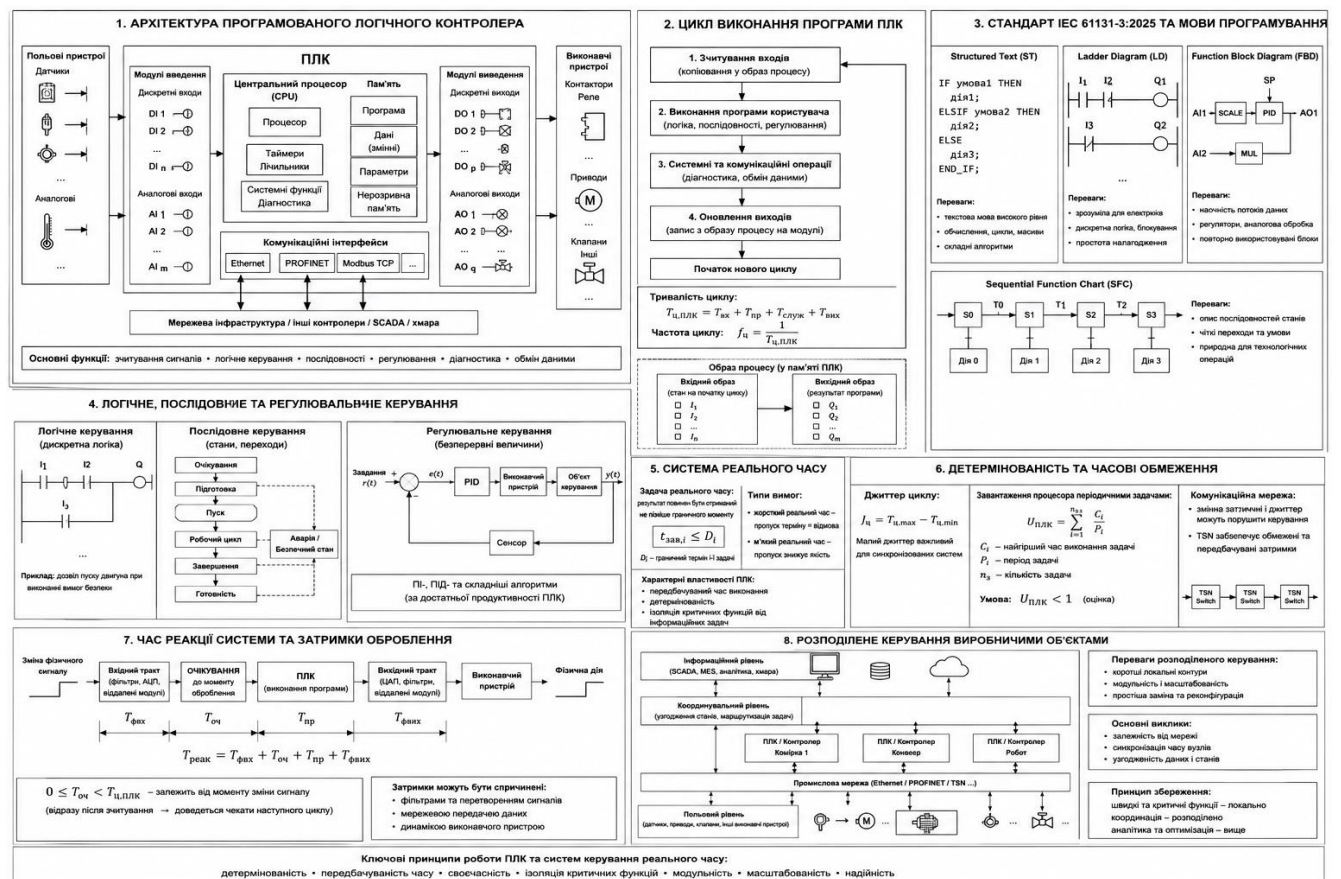


Рисунок 6.1 – Програмовані логічні контролери та системи керування реального часу

6.1. Архітектура програмованого логічного контролера

Програмований логічний контролер — ПЛК — є спеціалізованим промисловим обчислювальним пристроєм, призначеним для керування технологічним обладнанням у реальному часі. На відміну від універсального комп'ютера, його структура, операційне середовище та

інтерфейси орієнтовані на передбачуване циклічне або подієве виконання керувальних програм.

Типовий ПЛК містить:

- центральний процесор;
- оперативну та енергонезалежну пам'ять;
- модулі дискретного й аналогового введення-виведення;
- комунікаційні інтерфейси;
- системні таймери та лічильники;
- засоби діагностування;
- джерело живлення;
- системне програмне забезпечення.

Центральний процесор виконує програму користувача, системні функції та обмін даними. Пам'ять містить код програми, параметри, проміжні результати, технологічні змінні та діагностичну інформацію.

Вхідні модулі перетворюють сигнали польових пристроїв у внутрішнє цифрове представлення. Вихідні модулі виконують зворотнє перетворення і передають команди контакторам, приводам, клапанам та іншим виконавчим пристроям.

У традиційній архітектурі широко використовується **образ процесу**. На початку або в заданій фазі циклу контролер копіює фізичні входи у внутрішню область пам'яті. Програма працює з цією копією, а результати записуються в область виходів. Наприкінці циклу вони передаються фізичним вихідним модулям.

Це створює важливу властивість: протягом одного програмного циклу логічне представлення вхідного стану може залишатися незмінним, навіть якщо фізичний сигнал змінився під час виконання програми. Такий підхід спрощує аналіз послідовності виконання, але створює часову дискретність реакції.

Сучасні ПЛК виходять за межі класичної схеми «процесор — входи — виходи». Вони можуть виконувати локальне зберігання даних, протокольне перетворення, edge-функції та взаємодіяти з інформаційними платформами. У роботі J. Mellado та F. Núñez показано один із варіантів розширення ПЛК контейнеризованими функціями та засобами ІоТ при збереженні низькорівневого регулювання [3].

Проте додавання інформаційних функцій має обмеження. Обчислення, пов'язані з аналітикою або хмарною взаємодією, не повинні непередбачувано змінювати час виконання критичних алгоритмів керування.

6.2. Цикл виконання програми ПЛК

Для класичного ПЛК характерне повторюване циклічне виконання програми. Один цикл зазвичай складається з декількох послідовних фаз:

зчитування входів → виконання програми користувача → системні та комунікаційні операції → оновлення виходів → початок нового циклу.

Тривалість циклу можна представити:

$$T_{\text{ц,ПЛК}} = T_{\text{вх}} + T_{\text{пр}} + T_{\text{служ}} + T_{\text{вих}},$$

де $T_{\text{ц,ПЛК}}$ — тривалість циклу ПЛК; $T_{\text{вх}}$ — час зчитування входів; $T_{\text{пр}}$ — час виконання програми користувача; $T_{\text{служ}}$ — час системних, діагностичних і комунікаційних операцій; $T_{\text{вих}}$ — час оновлення виходів.

Частота повторення циклу становить:

$$f_{\text{ц}} = \frac{1}{T_{\text{ц,ПЛК}}}$$

Час циклу не обов'язково є абсолютно постійним. Він може залежати від кількості виконаних гілок програми, активності комунікаційних функцій, використання системних сервісів та інших задач.

Тому при проектуванні слід оцінювати не тільки середній, а й максимальний час циклу.

Якщо технологічний сигнал змінюється значно швидше, ніж контролер виконує повний цикл, окремі події можуть бути пропущені. Для швидких імпульсів використовуються спеціальні швидкі входи, апаратні лічильники або переривання, які не залежать безпосередньо від звичайного циклічного опитування.

Звідси випливає принцип: період виконання керувальної задачі повинен визначатися динамікою фізичного процесу, а не лише можливостями процесора.

У сучасному ПЛК одночасно можуть існувати задачі з різними періодами. Наприклад, регулювання електропривода виконується швидко, логіка виробничої операції — повільніше, а формування діагностичного звіту — ще рідше. Такий поділ дозволяє не витрачати процесорний ресурс на однаково часте виконання всіх функцій.

6.3. Стандарт ІЕС 61131-3 та мови програмування контролерів

ІЕС 61131-3 є базовим міжнародним стандартом для мов програмування ПЛК. Вона визначає синтаксис і семантику уніфікованого набору мов програмування контролерів [1].

У чинній редакції основними мовами є:

- Structured Text — ST;

- Ladder Diagram — LD;
- Function Block Diagram — FBD.

Для структуризації послідовного виконання програм і функціональних блоків застосовуються елементи Sequential Function Chart — SFC [1].

Structured Text

ST є текстовою мовою високого рівня. За структурою вона придатна для реалізації математичних обчислень, умов, циклів, оброблення масивів і складних алгоритмів.

Умовна конструкція може мати вигляд:

IF умова THEN дія ELSE альтернативна дія.

ST зручна тоді, коли алгоритм містить числові обчислення або значну кількість взаємопов'язаних умов.

Ladder Diagram

LD походить від релейно-контактних схем. Програма графічно представляється у вигляді контактів, котушок та функціональних елементів.

Цей формат зручний для дискретної логіки, блокувань, пускових схем та обслуговування обладнання персоналом, який звик працювати з електричними схемами.

Function Block Diagram

FBD представляє алгоритм як мережу взаємопов'язаних функціональних блоків. Такий підхід добре відповідає регуляторам, аналоговій обробці сигналів та повторно використовуваним функціям.

Sequential Function Chart

SFC дозволяє описувати процес як послідовність кроків і переходів. Це природно для виробничих операцій: завантаження → затиск → оброблення → розвантаження.

При цьому вибір мови не повинен визначатися лише особистою звичкою програміста. Логічну міжблокувальну схему часто простіше перевірити в LD, складні математичні обчислення — у ST, а послідовність станів — за допомогою SFC.

Дослідження практики промислового керування показують, що реальні програмні структури ПЛК значною мірою спираються на ІЕС 61131-3, але конкретні реалізації виробників можуть відрізнятися деталями виконання та системними можливостями [2].

Тому проєктна документація повинна містити не тільки назву мови, а й конкретне програмне середовище, версію системи виконання і прийнятий спосіб організації задач.

6.4. Логічне, послідовне та регулювальне керування

Програма виробничого ПЛК рідко складається лише з одного типу алгоритмів. У ній одночасно можуть виконуватися логічні блокування, технологічні послідовності та цифрове регулювання.

Логічне керування визначає стан виходу за комбінацією дискретних умов. Типовим прикладом є дозвіл запуску двигуна тільки при закритому захисному огороженні, відсутності аварії та готовності привода.

Послідовне керування визначає порядок переходу між технологічними станами. Тут важливим є не тільки поточний набір сигналів, а й попередній стан системи.

Наприклад:

очікування → підготовка → пуск → робочий цикл → завершення → готовність.

За аварійної події з будь-якого робочого стану може здійснюватися перехід до безпечного режиму.

Регулювальне керування працює з безперервними фізичними величинами: температурою, тиском, швидкістю, рівнем та витратою. На ПЛК реалізуються ПІ-, ПД- та, за достатньої обчислювальної продуктивності, складніші алгоритми.

Можливість реалізації складних алгоритмів керування в промислових ПЛК продемонстрована, зокрема, для модельно-прогнозного керування в реальному часі [4]. Проте така реалізація потребує перевірки часу виконання оптимізаційного алгоритму для найгіршого випадку, а не лише його середньої продуктивності.

Правильна програмна структура повинна розділяти ці функції. Наприклад, несправність аналітичного алгоритму не повинна відключати базову аварійну логіку. Аналогічно зупинка технологічної послідовності не повинна порушувати роботу незалежної системи безпеки.

Такий поділ зменшує приховані залежності між програмними блоками та спрощує тестування.

6.5. Поняття системи реального часу

Термін «**реальний час**» часто помилково ототожнюють із максимально високою швидкістю обчислень. Насправді система реального часу визначається часовою передбачуваністю.

Результат обчислення повинен бути не тільки правильним, а й отриманим до моменту, після якого він втрачає технологічний зміст.

Для окремої задачі необхідна умова:

$$t_{\text{зав},i} \leq D_i,$$

де $t_{\text{зав}}$ — момент завершення відповідної задачі; D_i — її граничний допустимий момент виконання.

Залежно від наслідків порушення часової межі розрізняють жорсткі та м'які вимоги реального часу.

У **жорсткій системі реального часу** пропуск граничного моменту розглядається як відмова функції. Прикладом є своєчасне формування команди у швидкому контурі руху або критичній технологічній операції.

У **м'якій системі реального часу** окреме запізнення знижує якість, але не обов'язково створює небезпечний стан. До цієї категорії може належати оновлення частини диспетчерської інформації.

ПЛК проектується насамперед як платформа для передбачуваного виконання керувальних задач. Саме часові гарантії відрізняють його від універсального комп'ютера, де операційна система може перерозподіляти ресурси відповідно до великої кількості непередбачуваних фонових процесів.

При цьому розвиток Industry 4.0 створює суперечність. Від ПЛК одночасно очікується класична детермінована поведінка і розширена інтеграція з IT-сервісами. Дослідження IoT-PLC показують можливі способи поєднання цих функцій, але низькорівневе керування все одно потребує захисту від неконтрольованого впливу інформаційних задач [3].

6.6. Детермінованість і часові обмеження

Детермінованість означає можливість визначити або обмежити час виконання операції з необхідною точністю.

Для технологічного керування недостатньо знати, що програма «зазвичай» виконується за 5 мс. Необхідно встановити, наскільки може змінюватися цей час при максимальному навантаженні.

Розкид тривалості циклу можна характеризувати:

$$J_{\text{ц}} = T_{\text{ц,max}} - T_{\text{ц,min}},$$

де $J_{\text{ц}}$ — джиттер циклу; $T_{\text{ц,max}}$ — максимальна зафіксована або гарантована тривалість циклу; $T_{\text{ц,min}}$ — мінімальна тривалість циклу.

Малий джиттер особливо потрібний у синхронізованих системах руху та розподіленому регулюванні. Якщо однакові команди формуються через неоднакові інтервали, це фактично створює змінний період дискретизації.

Для декількох періодичних задач оцінюють завантаження процесора:

$$U_{\text{ПЛК}} = \sum_{i=1}^{n_3} \frac{C_i}{P_i},$$

де $U_{\text{ПЛК}}$ — розрахункове завантаження процесора періодичними задачами; n_z — кількість задач; C_i — найгірший час виконання задачі; P_i — її період запуску.

Саме використання найгіршого часу виконання тут принципове. Середній час не гарантує, що окрема ітерація не порушить граничний термін.

Умова $U_{\text{ПЛК}} < 1$ означає лише, що сумарна обчислювальна потреба не перевищує доступний час у простій оцінці. Вона не є універсальним доказом планованості, оскільки результат залежить від пріоритетів, блокувань, взаємодії задач і способу планування.

Детермінованість потрібна не тільки процесору, а й комунікаційній системі. Звичайний мережевий обмін без гарантій може мати змінні затримки. Для промислових систем розробляються технології Time-Sensitive Networking, які використовують синхронізацію, планування трафіку та інші механізми для передбачуваного передавання часово-критичних даних [5].

6.7. Час реакції системи та затримки оброблення

Час циклу ПЛК і час реакції системи — не одне й те саме.

Фізичний сигнал спочатку проходить через вхідний тракт. Далі контролер повинен його зчитати, виконати відповідну частину програми і передати результат до вихідного модуля.

Загальний час реакції:

$$T_{\text{реак}} = T_{\text{фвх}} + T_{\text{оч}} + T_{\text{пр}} + T_{\text{фвих}},$$

де $T_{\text{реак}}$ — час від зміни фізичного входу до появи відповідної фізичної вихідної дії; $T_{\text{фвх}}$ — затримка вхідного тракту; $T_{\text{оч}}$ — очікування до моменту оброблення події; $T_{\text{фвих}}$ — затримка вихідного тракту.

У простій циклічній системі очікування залежить від моменту, коли змінився вхід:

$$0 \leq T_{\text{оч}} < T_{\text{ц,ПЛК}}.$$

Якщо сигнал змінюється безпосередньо перед зчитуванням входів, він може бути оброблений майже відразу. Якщо зміна відбулася одразу після відповідної фази, доведеться чекати наступного циклу.

Реальний час реакції може бути більшим через фільтри входів, віддалені модулі введення-виведення, мережеву передачу, черги повідомлень і динаміку самого виконавчого механізму.

Через це при проєктуванні необхідно аналізувати повний шлях сигналу, а не лише паспортний час циклу центрального процесора.

Для розподілених систем комунікаційна затримка стає частиною контуру керування. Дослідження TSN показують, що для промислових часово-критичних задач важливими є не тільки мала середня затримка, а її обмеженість і малий джиттер [5].

Отже, мережа з високою номінальною швидкістю передавання не обов'язково краща для керування, якщо час доставки окремого повідомлення суттєво змінюється.

6.8. Розподілене керування виробничими об'єктами

У простій автоматизованій машині всі функції можуть виконуватися одним ПЛК. При збільшенні масштабу кількість сигналів, приводів та технологічних модулів зростає, і централізований підхід стає менш зручним.

Розподілене керування передбачає, що окремі функції виконуються декількома контролерами або інтелектуальними виробничими модулями.

Наприклад:

- ПЛК технологічної комірки керує локальними приводами;
- контролер конвеєра забезпечує транспортування;
- контролер робота виконує траєкторне керування;
- координувальний рівень узгоджує їхні стани;
- інформаційний рівень отримує результати виконання.

Таке розділення скорочує довжину критичних локальних контурів і підвищує модульність системи. Якщо одна виробнича комірка має автономну логіку, її простіше замінити або реконфігурувати без переписування всієї системи.

Водночас розподіленість створює залежність від комунікацій. Два контролери можуть мати правильні локальні алгоритми, але працювати неправильно разом через запізнення, втрату повідомлення або різне трактування станів.

Особливо важлива синхронізація часу. Різницю між часовими шкалами двох вузлів можна оцінити:

$$\Delta t_{ab} = |t_a - t_b|,$$

де Δt_{ab} — розбіжність часових шкал вузлів a і b ; t_a та t_b — їхні локальні часові мітки тієї самої події.

Для звичайного виробничого обліку невелика розбіжність може бути несуттєвою. Для координованого руху або зіставлення швидких вимірювань вона може зробити дані некоректними.

Розподілене керування не означає, що всі рішення потрібно децентралізувати. Порівняльні дослідження централізованих і

розподілених підходів показують, що локальні рішення можуть забезпечувати швидшу реакцію та автономність, тоді як централізований рівень має перевагу при глобальній координації [6].

Тому для КФВС доцільна багаторівнева структура. Аварійні функції та швидкі регулятори виконуються локально. Взаємне узгодження виробничих модулів здійснюється розподілено або на координувальному рівні. Планування, аналіз та оптимізація можуть виконуватися вище.

Огляд промислових CPS також показує, що ПЛК залишаються важливою частиною фізичного рівня керування навіть при переході до складніших цифрових і розподілених виробничих архітектур [7].

Висновки до лекції 6

Програмований логічний контролер є базовим обчислювальним компонентом локального виробничого керування. Його архітектура орієнтована не лише на виконання програм, а на передбачувану взаємодію з фізичними входами та виходами.

Циклічний принцип роботи створює дискретний часовий характер реакції ПЛК. Тому для кожної функції необхідно аналізувати тривалість циклу, момент зчитування входів, час виконання програми та оновлення виходів.

ІЕС 61131-3:2025 визначає сучасний набір мов програмування контролерів — ST, LD та FBD, а також елементи SFC для структуризації послідовного виконання. Вибір конкретного представлення повинен відповідати природі алгоритму, а не лише звичці програміста.

Система реального часу відрізняється від просто швидкої системи наявністю часових гарантій. Для критичної задачі необхідно знати не середній, а максимально можливий час її виконання. Детермінованість передбачає обмежений час реакції та контроль джиттера.

У розподіленій КФВС часові властивості визначаються вже не одним ПЛК, а повним трактом від сенсора через контролер і мережу до виконавчого механізму. Тому комунікаційні затримки і синхронізація часу стають частиною задачі автоматичного керування.

Рациональна архітектура поєднує локальну автономність і координацію: швидкі та критичні функції залишаються поблизу фізичного об'єкта, а інформаційні й оптимізаційні функції розподіляються між вищими рівнями КФВС.

Контрольні питання

1. Які основні компоненти входять до архітектури ПЛК?
2. Що розуміють під образом процесу?
3. Які основні етапи утворюють цикл виконання ПЛК?
4. Чому максимальний час циклу важливіший за середній для критичних задач?
5. Які мови визначає актуальна редакція ІЕС 61131-3?
6. Для яких алгоритмів доцільно використовувати ST?
7. Для яких задач зручний LD?
8. Яке призначення SFC?
9. Чим логічне керування відрізняється від послідовного?
10. Що означає робота системи в реальному часі?
11. Чим детермінованість відрізняється від просто високої швидкодії?
12. Що характеризує джиттер циклу?
13. Від яких складових залежить час реакції ПЛК?
14. Чому розподілена система потребує синхронізації часу?
15. Які функції доцільно залишати локальними при побудові розподіленої КФВС?

Список використаних джерел

1. IEC 61131-3:2025. Programmable controllers — Part 3: Programming languages. 4th ed. Geneva : International Electrotechnical Commission, 2025.
2. Witte M., Sehr M. A., Ugalde I., Neidig J., Niknami M., Hoeme S., Lee E. A. Opportunities for Industrial Control. *IFAC-PapersOnLine*. 2020. Vol. 53, No. 2. P. 7839–7844. DOI: 10.1016/j.ifacol.2020.12.1951.
3. Mellado J., Núñez F. Design of an IoT-PLC: A containerized programmable logical controller for the Industry 4.0. *Journal of Industrial Information Integration*. 2022. Vol. 25. Article 100250. DOI: 10.1016/j.jii.2021.100250.
4. Krupa P., Saraf N., Limón D., Bemporad A. PLC implementation of a real-time embedded MPC algorithm based on linear input/output models. *IFAC-PapersOnLine*. 2020. Vol. 53, No. 2. P. 6987–6992. DOI: 10.1016/j.ifacol.2020.12.424.
5. Fedullo T., Morato A., Tramarin F., Rovati L., Vitturi S. A Comprehensive Review on Time Sensitive Networks with a Special Focus on Its Applicability to Industrial Smart and Distributed Measurement Systems. *Sensors*. 2022. Vol. 22, No. 4. Article 1638. DOI: 10.3390/s22041638.
6. Antons O., Arlinghaus J. C. Adaptive self-learning distributed and centralized control approaches for smart factories. *Procedia CIRP*. 2021. Vol. 104. P. 1577–1582. DOI: 10.1016/j.procir.2021.11.266.
7. Oks S. J., Jalowski M., Lechner M. et al. Cyber-Physical Systems in the Context of Industry 4.0: A Review, Categorization and Outlook. *Information Systems Frontiers*. 2024. Vol. 26. P. 1731–1772. DOI: 10.1007/s10796-022-10252-x.

ЛЕКЦІЯ 7. ПРОМИСЛОВІ МЕРЕЖІ ТА ТЕХНОЛОГІЇ ОБМІНУ ДАНИМИ

Мета лекції

Мета лекції полягає у формуванні уявлення про принципи побудови промислових комунікаційних мереж у складі кіберфізичних виробничих систем, їх часові характеристики, топологію та способи організації інформаційного обміну. Розглядаються промисловий Ethernet, Modbus TCP, PROFINET, EtherCAT, OPC UA, MQTT, технології детермінованого передавання даних, синхронізація часу та критерії вибору мережі для конкретного виробничого процесу.

У кіберфізичній виробничій системі мережа не є лише засобом передавання інформації між комп'ютерами. Вона може входити безпосередньо до контуру керування: через неї передаються вимірювальні сигнали, завдання приводам, команди контролерів, стани обладнання та повідомлення про аварії. Через це затримка, джиттер, пропускну здатність і надійність мережі можуть впливати на поведінку фізичного об'єкта [1, 2].

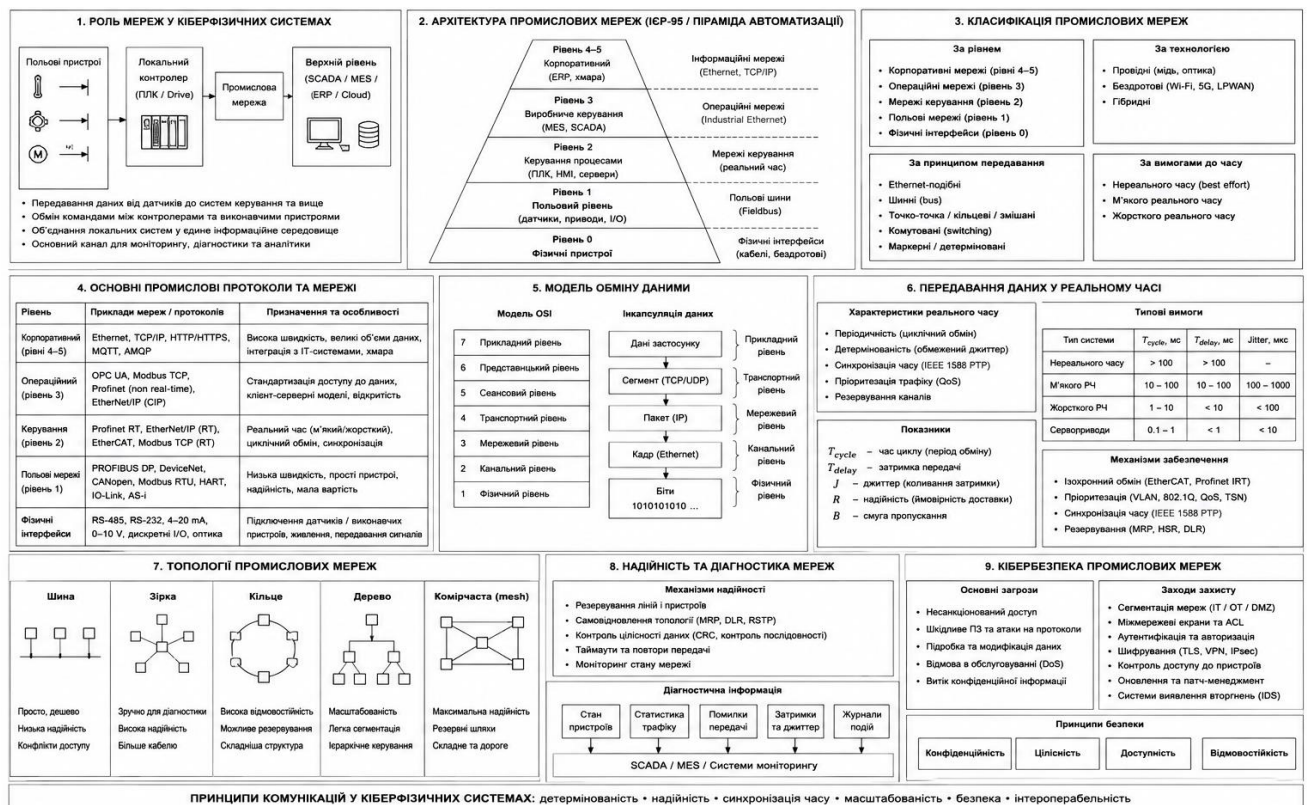


Рисунок 7.1 - Промислові мережі та технології обміну даними

7.1. Комунікаційна підсистема кіберфізичного виробництва

Комунікаційна підсистема КФВС забезпечує обмін даними між датчиками, виконавчими пристроями, ПЛК, приводами, периферійними

обчислювальними вузлами, SCADA, MES та іншими інформаційними системами.

У межах одного виробництва одночасно можуть передаватися різні за призначенням потоки:

- циклічні вимірювання;
- керувальні команди;
- сигнали синхронізації;
- аварійні повідомлення;
- параметри налаштування;
- діагностична інформація;
- технологічні рецепти;
- історичні дані;
- повідомлення аналітичних систем.

Для цих потоків не можна встановити однакові вимоги. Наприклад, архівний запис температури може бути переданий із затримкою в секунди без помітного впливу на технологічний процес. Для координованого керування кількома сервоприводами затримки та їх варіації повинні бути значно меншими і передбачуваними.

Час безпосереднього передавання пакета по каналу можна оцінити:

$$T_{\text{пак}} = \frac{L_{\text{пак}}}{C_{\text{л}}},$$

де $T_{\text{пак}}$ — час передавання пакета; $L_{\text{пак}}$ — довжина пакета, біт; $C_{\text{л}}$ — швидкість передавання каналу, біт/с.

Це співвідношення характеризує лише час серіалізації даних. Реальне повідомлення додатково проходить через мережеві інтерфейси, комутатори, програмні стеки й буфери. Тому фактична затримка більша.

Для виробничого керування суттєва не тільки середня затримка, а й можливість визначити її граничне значення. Саме необхідність передбачуваного передавання даних стала однією з причин розвитку промислових Ethernet-технологій та Time-Sensitive Networking [2, 7].

7.2. Топології промислових мереж

Топологія визначає спосіб фізичного або логічного з'єднання мережевих компонентів. У виробничих системах застосовуються лінійна, зіркоподібна, кільцева, деревоподібна та комбінована топології.

Лінійна топологія

У лінійній структурі вузли послідовно з'єднані один з одним. Така схема зручна для конвеєрних ліній, транспортних систем та обладнання, розташованого вздовж технологічного маршруту.

Її перевага — відносно невелика довжина кабельної інфраструктури. Недолік — можливість втрати зв'язку з частиною вузлів при пошкодженні окремої ділянки, якщо не передбачено резервний маршрут.

Топологія «зірка»

Усі кінцеві пристрої підключаються до центрального комутатора.

Така структура спрощує діагностування, оскільки пошкодження одного кабелю зазвичай не порушує роботу інших сегментів. Водночас сам центральний комутатор стає критичним компонентом.

Кільцева топологія

Кільце дозволяє створити два напрямки передавання даних. За наявності відповідного протоколу резервування розрив одного сегмента не обов'язково призводить до повної втрати зв'язку.

Деревоподібна структура

Дерево доцільне для великих систем з ієрархічно організованими виробничими ділянками. Воно добре масштабується, але потребує контролю навантаження на верхні вузли.

Повну затримку передачі через кілька мережевих вузлів можна представити:

$$T_{\text{мер}} = \sum_{h=1}^H (T_{\text{пак},h} + T_{\text{обр},h} + T_{\text{черг},h}),$$

де $T_{\text{мер}}$ — загальна мережева затримка; H — кількість послідовних ділянок; h — номер ділянки; $T_{\text{обр},h}$ — час оброблення пакета; $T_{\text{черг},h}$ — час його очікування в черзі.

Збільшення кількості проміжних вузлів може збільшувати як саму затримку, так і її варіацію. Тому топологія повинна оцінюватися разом із часовими вимогами.

7.3. Ethernet-технології у виробничих системах

Ethernet став основою для значної частини сучасних промислових мереж. Його перевагою є уніфікована фізична інфраструктура, велика пропускна здатність і сумісність із технологіями TCP/IP.

Звичайний Ethernet, однак, не створювався як спеціалізована система жорсткого реального часу. При одночасному передаванні багатьох потоків пакети можуть очікувати в чергах, а фактична затримка — змінюватися.

На базі Ethernet були створені спеціалізовані промислові технології:

- PROFINET;
- EtherCAT;
- EtherNet/IP;

- POWERLINK;
- SERCOS III та інші.

Вони використовують різні механізми для забезпечення циклічного обміну, пріоритетів, синхронізації та часової передбачуваності.

PROFINET орієнтований на широкий спектр задач — від параметризації та діагностичного обміну до циклічного передавання сигналів і керування рухом [3]. Офіційний опис PI вказує, що через одну Ethernet-інфраструктуру можуть одночасно виконуватися різні класи комунікацій.

Для сучасної КФВС істотним є одночасне існування двох груп трафіку.

ОТ-трафік пов'язаний із фізичним керуванням: сигналами ПЛК, приводів, датчиків і виконавчих пристроїв.

ІТ-трафік містить діагностику, файли, виробничі дані, аналітику та обмін із серверними системами.

Механізми Time-Sensitive Networking розвивають Ethernet для одночасного передавання таких потоків із різними вимогами до часу. TSN включає групу стандартів IEEE, пов'язаних із синхронізацією, плануванням трафіку, резервуванням та управлінням ресурсами [7].

7.4. Modbus TCP, PROFINET та EtherCAT

Промислові Ethernet-протоколи мають різне призначення. Їх не слід оцінювати за принципом «кращий — гірший» без прив'язки до технологічної задачі.

Modbus TCP

Modbus TCP використовується для обміну технологічними змінними через TCP/IP. Типовим є обмін у режимі запит–відповідь.

Він зручний для:

- вимірювальних пристроїв;
- регуляторів;
- перетворювачів;
- ПЛК;
- SCADA;
- допоміжного обладнання.

Основною перевагою є простота організації обміну та велика кількість сумісних пристроїв.

Для послідовного опитування декількох пристроїв:

$$T_{\text{опит}} = \sum_{j=1}^{n_{\text{оп}}} (T_{\text{зап},j} + T_{\text{відп},j}),$$

де $T_{\text{опит}}$ — тривалість циклу опитування; $n_{\text{оп}}$ — кількість пристроїв; $T_{\text{зап},j}$ — час виконання запиту; $T_{\text{відп},j}$ — час отримання відповіді.

При збільшенні кількості пристроїв або обсягу даних цикл опитування зростає. Тому Modbus TCP може бути цілком придатним для повільних технологічних параметрів, але не є природним вибором для високоточної синхронізації приводів.

PROFINET

PROFINET є технологією Industrial Ethernet для промислової автоматизації. Вона підтримує циклічний обмін між контролером та польовими пристроями, діагностику, параметризацію та задачі керування рухом [3].

Офіційні матеріали PROFIBUS & PROFINET International описують PROFINET як технологію, що дозволяє через Ethernet обслуговувати як звичайний TCP/IP-обмін, так і часово-критичні виробничі комунікації [3].

Така властивість корисна для КФВС, оскільки одна фізична інфраструктура може об'єднувати різні рівні автоматизації.

EtherCAT

EtherCAT використовує спеціальний принцип оброблення Ethernet-кадру «на проході». Повідомлення проходить через послідовність вузлів, а кожен пристрій отримує або записує призначені йому дані без формування окремого повного циклу Ethernet-обміну для кожного вузла [4].

Це дозволяє зменшити накладні витрати при циклічному обміні з великою кількістю пристроїв.

EtherCAT також використовує механізм Distributed Clocks. Він забезпечує узгодження локальних часових шкал різних вузлів і потрібний, наприклад, для синхронної роботи кількох сервоприводів [4].

Отже, вибір між Modbus TCP, PROFINET та EtherCAT визначається не лише швидкістю Ethernet. Необхідно враховувати модель обміну, кількість пристроїв, необхідність синхронізації та допустимий час реакції.

7.5. OPC UA як засіб інформаційної інтеграції

OPC Unified Architecture — OPC UA — займає окреме місце серед промислових комунікаційних технологій. Його призначення виходить за межі простого передавання числового значення.

OPC UA дозволяє формувати **інформаційну модель об'єкта**: задавати обладнання, параметри, структуру, типи даних, одиниці вимірювання та зв'язки між компонентами.

Наприклад, замість передавання абстрактного значення «1480» система може визначити, що це частота обертання конкретного

електродвигуна, вказати одиницю вимірювання та положення цього двигуна в структурі установки.

Ця властивість має принципове значення для інтероперабельності КФВС.

OPC UA підтримує два різні підходи до взаємодії.

Client–Server

Клієнт звертається до сервера та виконує читання, запис або іншу операцію.

Такий підхід придатний для конфігурування, діагностування, доступу до параметрів і багатьох систем SCADA.

Publish–Subscribe

У PubSub видавець формує повідомлення, а підписники отримують потрібні їм дані без прямого встановлення індивідуального зв'язку між кожною парою компонентів.

Актуальна специфікація OPC UA Part 14 визначає PubSub як модель, що доповнює Client–Server, а також передбачає різні транспортні відображення, зокрема UDP та брокерні протоколи [5].

Структурно:

Publisher → комунікаційне середовище → Subscriber.

При цьому кількість підписників не повинна вимагати створення окремого механізму передавання для кожного з них. Це корисно для масштабованих систем збору виробничих даних [5].

OPC UA може застосовуватися разом із TSN. У такому випадку OPC UA вирішує задачі інформаційної моделі та прикладного обміну, а TSN забезпечує властивості нижчих мережеских рівнів, необхідні для детермінованого транспорту [7, 8].

7.6. MQTT та обмін повідомленнями в IoT-системах

MQTT використовується переважно для обміну даними між розподіленими IoT-компонентами. Він побудований за моделлю publish/subscribe.

Типова структура:

видавець → брокер → підписник.

Видавець надсилає повідомлення не безпосередньо конкретному одержувачу, а у визначену тему. Брокер приймає повідомлення і розподіляє його між підписниками.

Такий підхід послаблює залежність між компонентами.

Наприклад, сенсорний шлюз може передавати виробничі дані брокеру. Надалі до цього потоку можуть незалежно підключитися:

- система моніторингу;
- база даних;
- аналітичний сервіс;
- система прогнозування;
- хмарна платформа.

При цьому програму самого сенсорного шлюзу не потрібно змінювати для кожного нового споживача.

MQTT характеризується відносно малими накладними витратами та зручний для телеметрії й ІоТ-архітектур [6].

Проте брокерна модель має архітектурне обмеження. Якщо критичний потік залежить від одного брокера, його відмова може порушити обмін між великою кількістю компонентів. Тому для відповідальних систем застосовується резервування брокерів або інша відмовостійка структура.

MQTT не слід автоматично використовувати для швидких локальних контурів керування. Його природна область — передавання виробничих даних між ОТ-, edge- та ІТ-рівнями, а не безпосереднє високодинамічне регулювання.

7.7. Затримка, джиттер, пропускна здатність і втрати пакетів

Для оцінювання промислової мережі недостатньо знати лише номінальну швидкість Ethernet.

Необхідно аналізувати:

- затримку;
- джиттер;
- завантаження;
- втрати;
- стабільність часу доставки.

Затримка визначає час проходження повідомлення від джерела до одержувача.

Джиттер характеризує зміну цієї затримки.

$$J_{\text{мер}} = D_{\text{max}} - D_{\text{min}},$$

де $J_{\text{мер}}$ — мережевий джиттер; D_{max} — максимальна затримка доставки; D_{min} — мінімальна затримка доставки.

Для архівної інформації невеликий джиттер часто не має значення. Для синхронного керування приводами він може призводити до різних моментів виконання команд.

Завантаження каналу можна оцінити:

$$\rho_{\text{мер}} = \frac{R_{\Sigma}}{C_{\text{л}}},$$

де $\rho_{\text{мер}}$ — коефіцієнт завантаження мережевого каналу; R_{Σ} — сумарний потік даних.

При наближенні завантаження до фізичної межі каналу зростає ймовірність утворення черг. Тому мережу реального часу не слід проєктувати з розрахунку на постійну роботу біля максимальної пропускної здатності.

Частку втрачених пакетів:

$$P_{\text{втр}} = \frac{N_{\text{втр}}}{N_{\text{перед}}},$$

де $P_{\text{втр}}$ — відносна частка втрат; $N_{\text{втр}}$ — кількість втрачених пакетів; $N_{\text{перед}}$ — кількість переданих пакетів.

Для інформаційної системи окремий втрачений пакет інколи можна повторити. У жорсткому контурі реального часу повторна передача може вже бути запізнілою. Тому потрібна не лише висока ймовірність доставки, а визначена поведінка системи при втраті повідомлення.

7.8. Синхронізація часу в розподілених КФВС

Кожний мережевий вузол має локальний генератор часу. Його частота не є абсолютно точною, тому годинники різних пристроїв поступово розходяться.

Для значної частини виробничих задач це потрібно компенсувати.

Синхронізація необхідна для:

- координованого керування рухом;
- порівняння сигналів різних сенсорів;
- визначення послідовності аварійних подій;
- розподілених вимірювань;
- роботи TSN;
- високоточного журналювання.

Часове зміщення вузла відносно еталонної шкали:

$$\varepsilon_{\text{ч},i} = t_i - t_{\text{ref}},$$

де $\varepsilon_{\text{ч},i}$ — похибка синхронізації відповідного вузла; t_{ref} — еталонна часова шкала.

Розглянемо два сенсори, що реєструють одну механічну подію. Якщо їх часові шкали зміщені, система може зробити помилковий висновок про те, який процес виник раніше.

Для високоточної мережевої синхронізації використовуються механізми на основі Precision Time Protocol, апаратні часові мітки та спеціалізовані механізми конкретних промислових технологій.

В EtherCAT для цього застосовуються Distributed Clocks [4]. У TSN спільна часова база використовується для планування передачі критичних потоків у визначених часових вікнах [7].

Отже, синхронізацію часу необхідно розглядати як частину системи керування, а не лише як засіб правильного відображення годинника.

7.9. Вибір мережевої технології для виробничого застосування

Вибір мережі повинен починатися з характеристик фізичного процесу.

Перший критерій — **час реакції**. Для повільного контролю температури та багатокоординатного сервопривода потрібні різні комунікаційні рішення.

Другий — **джиттер**. Для асинхронного моніторингу він може бути другорядним, тоді як для синхронізованого руху стає одним із основних показників.

Третій — **обсяг даних**. Дискретні датчики створюють невеликий потік, тоді як високочастотні вимірювання або машинний зір потребують значної пропускної здатності.

Четвертий — **кількість вузлів і топологія**.

П'ятий — **необхідність інформаційної інтероперабельності**.

Шостий — **відмовостійкість**.

Сьомий — **кібербезпека**.

Восьмий — **можливість подальшого масштабування**.

Для практичної КФВС не обов'язково використовувати один протокол на всіх рівнях. Рациональною може бути структура, у якій:

- EtherCAT або інша мережа реального часу обслуговує приводи;
- PROFINET використовується для комплексної автоматизації установки;
- Modbus TCP — для допоміжних пристроїв;
- OPC UA — для структурованої інтеграції;
- MQTT — для IoT та передачі даних на edge- або cloud-рівень.

Така система не є надмірно складною, якщо функціональні межі кожної технології визначені заздалегідь.

Основна помилка при проектуванні полягає у виборі протоколу лише за номінальною швидкістю передачі. Для КФВС важливіша відповідність комунікаційної технології фізичній задачі: дані повинні бути не просто доставлені, а доставлені **вчасно, однозначно та з необхідною надійністю**.

Висновки до лекції 7

Промислова комунікаційна мережа є функціональною частиною кіберфізичної виробничої системи. Якщо дані використовуються в замкненому контурі керування, мережева затримка і джиттер безпосередньо впливають на поведінку фізичного процесу.

Промислові Ethernet-технології відрізняються механізмами організації циклічного обміну та реального часу. Modbus TCP придатний для відносно простого обміну параметрами. PROFINET охоплює широкий діапазон задач автоматизації. EtherCAT орієнтований на швидкий циклічний обмін і синхронізацію розподілених вузлів.

OPC UA вирішує не лише транспортну, а й інформаційну задачу, дозволяючи формалізувати зміст і структуру виробничих даних. MQTT забезпечує слабкозв'язану модель publish/subscribe та зручний для IoT і взаємодії з edge- та cloud-рівнями.

Для систем реального часу потрібно оцінювати не тільки пропускну здатність, а повну затримку, джиттер, втрати повідомлень і точність синхронізації. Розвиток TSN спрямований на створення більш уніфікованого Ethernet-середовища, в якому часово-критичний і звичайний інформаційний трафік можуть співіснувати з визначеними характеристиками [7].

Вибір мережі має виконуватися за вимогами конкретного технологічного процесу. У складній КФВС доцільним є поєднання кількох протоколів, якщо кожний із них використовується на тому рівні, де його характеристики відповідають поставленій задачі.

Контрольні питання

1. Яке призначення комунікаційної підсистеми КФВС?
2. Які типи даних передаються промисловими мережами?
3. Від чого залежить повна мережева затримка?
4. Які топології використовуються у виробничих мережах?
5. Чому стандартний Ethernet не завжди достатній для жорсткого реального часу?
6. Яке призначення TSN?
7. Для яких задач використовується Modbus TCP?
8. Які особливості має PROFINET?
9. У чому полягає принцип обміну EtherCAT?
10. Для чого використовуються Distributed Clocks?
11. Яка роль OPC UA у КФВС?
12. Чим модель Client–Server відрізняється від PubSub?
13. Як організовано обмін даними в MQTT?
14. Що характеризує мережевий джиттер?

15. Чому велике завантаження каналу може збільшувати затримку?
16. Для чого виконується синхронізація часу?
17. Які параметри необхідно враховувати при виборі промислової мережі?

Список використаних джерел

1. Pivoto D. G. S., de Almeida L. F. F., Righi R. R., Rodrigues J. J. P. C., Lugli A. B., Alberti A. M. Cyber-physical systems architectures for industrial internet of things applications in Industry 4.0: A literature review. *Journal of Manufacturing Systems*. 2021. Vol. 58. P. 176–192. DOI: 10.1016/j.jmsy.2020.11.017.
2. Fedullo T., Morato A., Tramarin F., Rovati L., Vitturi S. A Comprehensive Review on Time Sensitive Networks with a Special Focus on Its Applicability to Industrial Smart and Distributed Measurement Systems. *Sensors*. 2022. Vol. 22, No. 4. Article 1638. DOI: 10.3390/s22041638.
3. PROFIBUS & PROFINET International. *PROFINET Technology and Application: System Description*. Version 2024.02. Karlsruhe : PI, 2024.
4. EtherCAT Technology Group. *EtherCAT Basics: EtherCAT Communication Principles*. Nuremberg : ETG, 2024.
5. OPC Foundation. *OPC Unified Architecture — Part 14: PubSub. OPC 10000-14*. Version 1.05.06. Scottsdale : OPC Foundation, 2025.
6. Akshatha P. S., Dilip Kumar S. M., Venugopal K. R. MQTT Implementations, Open Issues, and Challenges: A Detailed Comparison and Survey. *International Journal of Sensors, Wireless Communications and Control*. 2022. Vol. 12, No. 8. P. 553–576. DOI: 10.2174/2210327913666221216152446.
7. Zhang T., Wang G., Xue C., Wang J., Nixon M., Han S. Time-Sensitive Networking (TSN) for Industrial Automation: Current Advances and Future Directions. *ACM Computing Surveys*. 2025. Vol. 57, No. 2. Article 30. DOI: 10.1145/3695248.
8. Trifonov H., Heffernan D. OPC UA TSN: a next-generation network for Industry 4.0 and IIoT. *International Journal of Pervasive Computing and Communications*. 2023. Vol. 19, No. 3. P. 386–411.

ЛЕКЦІЯ 8. ПРОМИСЛОВИЙ ІНТЕРНЕТ РЕЧЕЙ ТА ПЕРИФЕРІЙНІ ОБЧИСЛЕННЯ

Мета лекції

Мета лекції полягає у формуванні уявлення про принципи побудови Industrial Internet of Things у складі кіберфізичних виробничих систем та способи розподілу обчислень між польовими пристроями, ПЛК, edge-, fog- і cloud-рівнями. Розглядаються структура IIoT, промислові шлюзи, ідентифікація пристроїв, периферійне оброблення даних, співвідношення

edge-, fog- і cloud computing, часові та ресурсні обмеження, а також критерії вибору місця виконання алгоритму.

ІоТ не слід розглядати як просте підключення промислового обладнання до Інтернету. Його практичний зміст полягає у формуванні керованого інформаційного середовища, в якому сенсори, приводи, контролери, обчислювальні вузли та інформаційні сервіси можуть обмінюватися даними про стан виробництва. Для промислових застосувань при цьому зберігаються вимоги, характерні для автоматизації: передбачуваний час реакції, надійність, контроль доступу та можливість локальної роботи при порушенні зовнішнього зв'язку [1, 2].

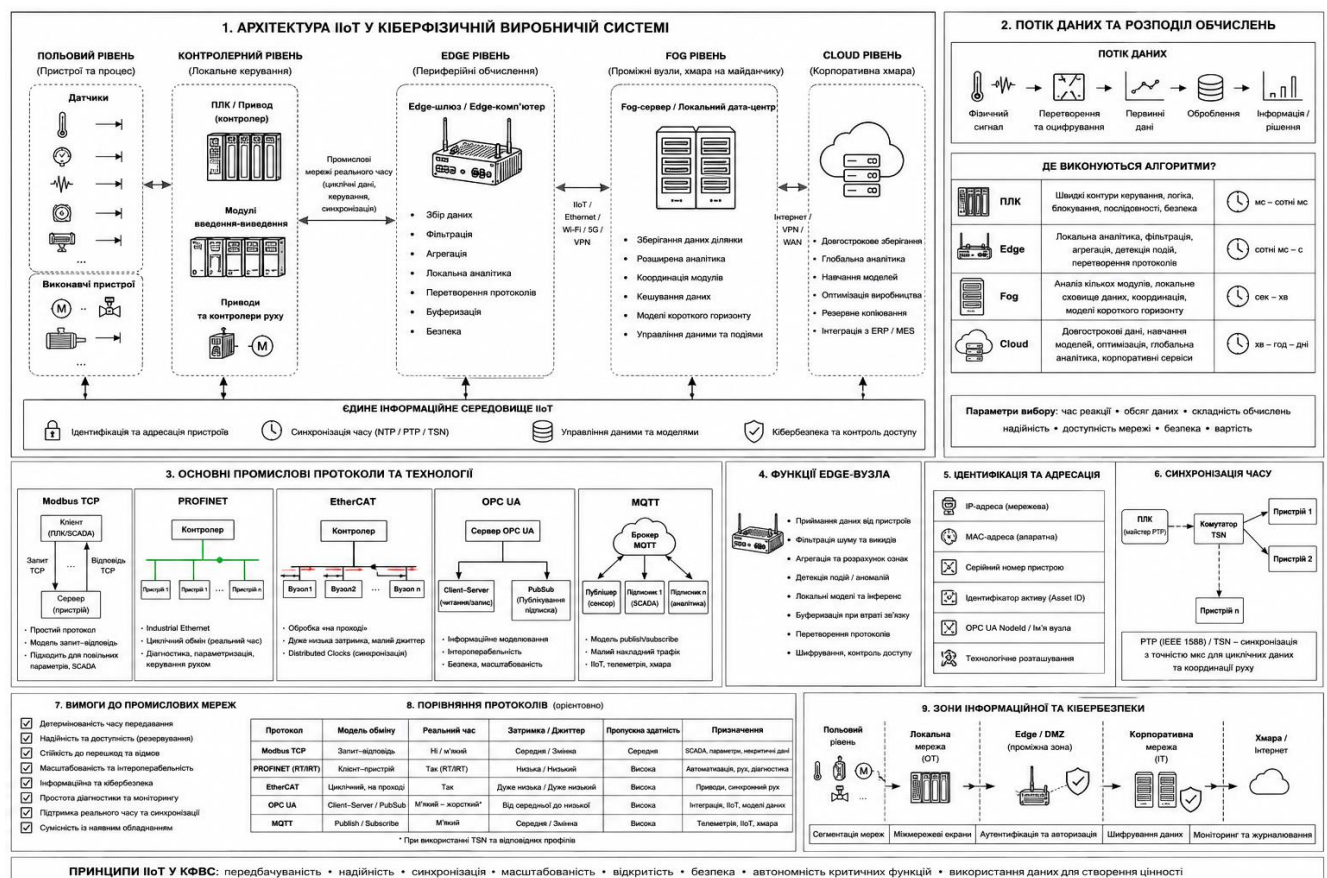


Рисунок 8.1 – Промисловий інтернет речей та периферійні обчислення

8.1. Концепція Industrial Internet of Things

Industrial Internet of Things — ІІОТ — являє собою сукупність підключених промислових пристроїв, програмних компонентів і комунікаційних засобів, які забезпечують отримання, передавання та використання даних про фізичні виробничі процеси.

До ІІОТ можуть входити:

- промислові сенсори;
- інтелектуальні виконавчі механізми;

- електроприводи;
- ПЛК;
- модулі віддаленого введення-виведення;
- промислові роботи;
- засоби технічного діагностування;
- edge-шлюзи;
- сервери;
- програмні аналітичні сервіси.

Від звичайного споживчого IoT промислове середовище відрізняється насамперед характером наслідків відмови. Запізнення інформації від побутового датчика часто лише погіршує зручність використання системи. У промисловому об'єкті неправильне або запізнile повідомлення може впливати на технологічний процес.

ПоТ тому тісно пов'язаний із КФВС. Кіберфізична система формує замкнений зв'язок між фізичним процесом і цифровим середовищем, а ПоТ забезпечує масштабовану інформаційну взаємодію між великою кількістю таких компонентів [1].

Виробничі пристрої можуть формувати значні потоки даних. Для однакових сенсорних каналів швидкість їх утворення можна оцінити:

$$G_{\text{дан}} = n_{\text{вуз}} b_{\text{відл}} f_{\text{відл}},$$

де $G_{\text{дан}}$ — швидкість формування первинних даних, біт/с; $n_{\text{вуз}}$ — кількість сенсорних вузлів; $b_{\text{відл}}$ — обсяг одного відліку, біт; $f_{\text{відл}}$ — частота формування відліків, Гц.

Якщо збільшується кількість сенсорів або частота вимірювання, обсяг інформації також зростає. Через це передавання всіх первинних даних без попереднього оброблення до віддаленого сервера не завжди раціональне.

Саме ця обставина стала однією з причин розвитку edge computing. Обчислення переміщуються ближче до джерела інформації, щоб частина рішень могла бути отримана без передавання всього потоку в центральну хмару [3, 4].

8.2. ПоТ-пристрої та шлюзи

ПоТ-пристрій повинен не тільки формувати дані, а й мати визначений спосіб їх передавання та опису. Найпростіший промисловий сенсор може не мати власного мережевого інтерфейсу. Він передає аналоговий або дискретний сигнал ПЛК, а вже контролер або шлюз включає ці дані до ПоТ-середовища.

Більш складний інтелектуальний пристрій може містити:

- сенсорний елемент;

- мікроконтролер;
- пам'ять;
- локальні алгоритми фільтрації;
- мережевий інтерфейс;
- засоби самодіагностування;
- параметри конфігурації.

Між польовим обладнанням та інформаційною мережею часто встановлюється **ПоТ-шлюз**.

Шлюз може виконувати декілька функцій:

- збирати дані з локальних пристроїв;
- перетворювати промислові протоколи;
- фільтрувати та агрегувати вимірювання;
- формувати часові мітки;
- виконувати локальні алгоритми;
- передавати дані на сервер або у хмару;
- контролювати доступ між мережевими сегментами.

Через це ПоТ-шлюз не слід ототожнювати зі звичайним мережевим маршрутизатором. У КФВС він часто виконує також обчислювальні та інформаційні функції.

Промисловий шлюз особливо потрібний під час модернізації існуючого обладнання. Наприклад, старий ПЛК може підтримувати тільки локальний промисловий протокол, тоді як система верхнього рівня використовує OPC UA або MQTT. Шлюз дозволяє інтегрувати ці рівні без заміни основного контролера.

Проте кожний додатковий шлюз створює новий програмний і мережевий компонент. Його відмова, неправильна конфігурація або перевантаження можуть порушити інформаційний обмін. Тому функції шлюзу повинні бути визначені ще на етапі побудови архітектури.

8.3. Ідентифікація та адресація виробничих об'єктів

Для роботи великої ПоТ-системи недостатньо передати повідомлення від одного вузла до іншого. Потрібно однозначно встановити, який саме фізичний або цифровий об'єкт є джерелом інформації.

Необхідно розрізняти поняття адреси та ідентифікатора.

Адреса визначає, куди потрібно доставити повідомлення в конкретній мережевій конфігурації.

Ідентифікатор відповідає на інше питання — який саме актив або компонент представлений у системі.

Наприклад, IP-адреса контролера може змінитися після реконфігурації мережі, але ідентичність технологічного об'єкта повинна залишитися незмінною.

У ПоТ застосовуються різні рівні ідентифікації:

- мережеві адреси;
- апаратні ідентифікатори;
- заводські серійні номери;
- програмні ідентифікатори;
- імена вузлів;
- ідентифікатори інформаційних моделей;
- ідентифікатори виробничих активів.

У мережах IPv6 теоретична кількість значень 128-бітного адресного простору становить:

$$\mathcal{N}_{\text{IPv6}} = 2^{128}.$$

Великий адресний простір полегшує підключення значної кількості мережевих компонентів, однак сам по собі не вирішує проблему семантичної ідентифікації.

Для цифрової моделі необхідно знати не тільки адресу пристрою, а й те, що він представляє: насос № 2, привід конвеєра, датчик температури конкретного підшипника або роботизовану комірку.

Тому ідентифікація ПоТ-пристроїв пов'язана з інформаційними моделями, які розглядалися в Лекції 7 при аналізі OPC UA.

8.4. Edge computing у виробничих системах

Edge computing означає виконання обчислень поблизу джерела даних — безпосередньо на пристрої, промисловому комп'ютері або периферійному вузлі виробничої мережі.

Головна причина такого підходу не в тому, що edge-пристрій обов'язково має більшу обчислювальну потужність. Навпаки, його ресурси часто обмежені порівняно з сервером. Перевага полягає в меншій відстані між фізичним процесом, даними та алгоритмом.

Систематичні огляди застосування edge computing у виробництві відзначають скорочення обсягу передавання даних і можливість отримувати результат ближче до місця виникнення події [3, 4].

На edge-рівні можуть виконуватися:

- цифрова фільтрація;
- агрегування даних;
- обчислення статистичних характеристик;
- виявлення аномалій;

- оброблення зображень;
- локальне машинне навчання або інференс;
- перетворення протоколів;
- формування діагностичних повідомлень.

Нехай edge-вузол отримує певний обсяг первинної інформації та передає після оброблення лише частину. Коефіцієнт збереженого потоку можна визначити:

$$k_{\text{ред}} = \frac{V_{\text{вих}}}{V_{\text{вх}}},$$

де $k_{\text{ред}}$ — коефіцієнт обсягу даних після периферійного оброблення; $V_{\text{вх}}$ — обсяг первинних даних; $V_{\text{вих}}$ — обсяг даних, переданих на наступний рівень.

Якщо $k_{\text{ред}} < 1$, edge-рівень зменшує обсяг трафіку.

Наприклад, вібраційний сенсор може формувати тисячі відліків за секунду. Для оперативного моніторингу не завжди потрібно постійно передавати весь часовий сигнал. Edge-вузол може розраховувати СКЗ, спектральні ознаки або індикатор аномалії, а первинні дані зберігати лише під час підозрілої події.

Проте локальне скорочення даних має бути обґрунтованим. Якщо первинна інформація видалена, повторно застосувати до неї інший метод аналізу вже неможливо.

8.5. Fog- та cloud-рівні оброблення даних

Edge-, fog- і cloud computing не є трьома назвами однієї технології. Вони відрізняються розташуванням обчислювальних ресурсів та характером задач.

Edge-рівень знаходиться максимально близько до фізичного джерела даних.

Fog computing формує проміжний розподілений обчислювальний рівень між локальними пристроями та централізованою хмарою.

Cloud computing надає значні централізовані або віртуалізовані обчислювальні та інформаційні ресурси.

Огляди fog computing показують, що його використання в IoT пов'язане з необхідністю скоротити затримки, розмістити частину зберігання та оброблення ближче до промислових пристроїв і не передавати весь потік безпосередньо у віддалений центр обробки даних [5].

При цьому fog computing не замінює хмару. Обидва рівні можуть використовуватися одночасно.

Хмара краще підходить для:

- довгострокового зберігання;
- порівняння даних кількох виробничих майданчиків;
- навчання ресурсомістких моделей;
- корпоративної аналітики;
- великих історичних вибірок;
- планових оптимізаційних задач.

Edge- та fog-рівні доцільніші там, де істотними є мала затримка, локальна автономність або обмеження мережевого трафіку.

Сучасні Edge-Fog-Cloud архітектури саме тому розглядаються як спільне обчислювальне середовище, в якому різні частини одного інформаційного процесу виконуються на різних рівнях [6].

8.6. Розподіл обчислювальних задач між ПЛК, edge- і cloud-рівнями

Проблема розподілу обчислень полягає не у виборі між «локально» та «у хмарі». У КФВС існує декілька рівнів, кожний з яких має власні переваги.

ПЛК

На ПЛК доцільно залишати:

- швидкі контури керування;
- логічні блокування;
- аварійні функції;
- технологічні послідовності;
- задачі, що повинні виконуватися при втраті зовнішньої мережі.

ПЛК має обмежені ресурси, але забезпечує передбачувану роботу безпосередньо біля фізичного об'єкта.

Edge-вузол

На edge-рівень доцільно переносити задачі, для яких потрібне швидке оброблення значних даних, але які не повинні безпосередньо виконувати функцію критичного ПЛК-керування:

- комп'ютерний зір;
- спектральний аналіз;
- локальне розпізнавання стану;
- агрегування;
- прогнозування короткого горизонту;
- локальна робота цифрової моделі.

Fog-рівень

Fog-вузол може координувати декілька edge-компонентів, накопичувати дані виробничої ділянки та виконувати задачі, для яких

локального edge-пристрою недостатньо, а передавання в хмару створює зайву затримку.

Cloud-рівень

Хмарне середовище має перевагу при великій потребі у зберіганні та обчисленнях, але його робота залежить від зовнішньої комунікаційної інфраструктури.

Тому аварійний захист верстата не повинен залежати від хмари, тоді як навчання моделі за даними декількох місяців експлуатації цілком може виконуватися віддалено.

Огляди промислового edge computing наголошують, що обмежені ресурси периферійних пристроїв змушують розподіляти обчислення відповідно до потрібної точності, затримки та складності алгоритму [3].

8.7. Затримки передавання та оброблення інформації

При виборі місця виконання алгоритму потрібно враховувати не тільки час самих обчислень.

Нехай задача виконується на рівні ℓ . Повний час отримання результату:

$$\Theta_{\ell} = \Theta_{\text{tx},\ell} + \Theta_{\text{wait},\ell} + \Theta_{\text{calc},\ell},$$

де Θ_{ℓ} — повний час отримання результату на вибраному рівні; $\Theta_{\text{tx},\ell}$ — сумарний час передавання даних і результату; $\Theta_{\text{wait},\ell}$ — час очікування доступу до обчислювального ресурсу; $\Theta_{\text{calc},\ell}$ — час виконання алгоритму.

У локальному edge-вузлі $\Theta_{\text{tx},\ell}$ зазвичай мала, але $\Theta_{\text{calc},\ell}$ може бути значною через обмежену продуктивність.

У хмарі ситуація може бути протилежною: потужний сервер швидко виконує складний алгоритм, але необхідно передати великий обсяг даних через декілька мереж.

Для грубої оцінки чистого часу обчислення:

$$\Theta_{\text{calc}} = \frac{n_{\text{op}}}{F_{\text{обч}}},$$

де n_{op} — орієнтовна кількість обчислювальних операцій; $F_{\text{обч}}$ — ефективна швидкість їх виконання.

Це спрощена залежність. Реальний час визначається архітектурою процесора, пам'яттю, паралельністю, операційною системою та програмною реалізацією.

Для задачі з часовим обмеженням повинна виконуватися умова:

$$\Theta_{\ell} \leq \Theta_{\text{гран}},$$

де $\Theta_{\text{гран}}$ — максимально допустимий час отримання результату.

Якщо умова не виконується, цей рівень обчислення непридатний для відповідної функції незалежно від інших переваг.

Саме затримка є однією з причин, через які централізоване cloud computing не може повністю замінити локальні та периферійні обчислення у виробничих системах [5, 6].

8.8. Вибір місця виконання алгоритмів керування й аналітики

Розміщення алгоритму повинно визначатися сукупністю критеріїв.

Перший критерій — **часова критичність**. Чим менший допустимий час реакції, тим ближче до фізичного об'єкта доцільно виконувати функцію.

Другий — **обчислювальна складність**. Великі моделі машинного навчання можуть перевищувати ресурси локального вузла.

Третій — **обсяг вихідних даних**. Якщо передавання первинного потоку створює значне мережеве навантаження, попередню обробку доцільно виконати локально.

Четвертий — **необхідність автономної роботи**. Об'єкт, який повинен залишатися працездатним при втраті зовнішнього зв'язку, не може залежати від хмарного сервісу у критичних функціях.

П'ятий — **конфіденційність та кібербезпека**. Передавання технологічних даних між OT-, edge- та cloud-рівнями розширює поверхню взаємодії і потребує додаткових захисних механізмів [7].

Шостий — **масштабованість**. Централізований сервер зручний для загальної координації, але зі збільшенням кількості потоків може стати вузьким місцем.

Для порівняння декількох варіантів розміщення обчислень можна використати узагальнений критерій:

$$\Psi_j = w_{\Theta} \Theta_j + w_{\text{ен}} E_j + w_{\text{тр}} B_j,$$

де Ψ_j — оцінка j -го варіанта розміщення; w_{Θ} — ваговий коефіцієнт часу; $w_{\text{ен}}$ — ваговий коефіцієнт енергетичних витрат; E_j — енергетичні витрати; $w_{\text{тр}}$ — ваговий коефіцієнт мережевого ресурсу; B_j — необхідний обсяг мережевого ресурсу.

Чим менше значення критерію, тим при прийнятих вагах кращий відповідний варіант.

Такий підхід є допоміжним. Якщо існує жорстка технологічна вимога, вона повинна перевірятися окремо. Наприклад, хмарний алгоритм може мати низькі витрати і високу обчислювальну точність, але він неприйнятний, якщо не гарантує потрібний час реакції.

Для виробничої КФВС найбільш обґрунтованою часто є не повністю локальна і не повністю хмарна структура, а ієрархічний розподіл задач:

ПЛК → edge → fog → cloud.

У цьому ланцюгу дані можуть поступово перетворюватися: від первинного сигналу до технологічної ознаки, від ознаки — до діагностичного стану, а від станів багатьох об'єктів — до виробничої аналітики.

Такий підхід дозволяє зберегти локальну швидкість і водночас використовувати значні серверні ресурси. Дослідження Edge–Fog–Cloud архітектур підтверджують доцільність саме спільного використання цих рівнів для різних типів ІоТ-даних та задач [6].

Висновки до лекції 8

Industrial Internet of Things формує інформаційну інфраструктуру, за допомогою якої виробничі пристрої, контролери, сенсори та обчислювальні сервіси включаються до спільного середовища обміну даними. Для промислових застосувань при цьому зберігаються вимоги до передбачуваного часу реакції, надійності й автономності локальних функцій.

ІоТ-шлюз виконує не тільки мережеве з'єднання. Він може здійснювати перетворення протоколів, агрегування, фільтрацію та первинну аналітику. Це дозволяє інтегрувати обладнання різних поколінь і скорочувати потоки даних, що передаються на верхні рівні.

Edge computing переносить частину обчислень ближче до фізичного процесу. Це зменшує мережеву складову затримки та дозволяє приймати частину рішень локально. Водночас edge-вузли мають обмежені обчислювальні й енергетичні ресурси, тому не всі задачі доцільно виконувати на периферії.

Fog computing утворює проміжний розподілений рівень, а cloud computing забезпечує значні централізовані ресурси для зберігання, складної аналітики та оброблення великих історичних масивів.

Вибір місця виконання алгоритму визначається часовими вимогами, обсягом даних, складністю обчислень, автономністю, мережевими ресурсами та кібербезпекою. Критичні контури керування повинні залишатися поблизу фізичного об'єкта, тоді як аналітичні задачі можуть розподілятися між edge-, fog- і cloud-рівнями.

Контрольні питання

1. Що розуміють під Industrial Internet of Things?
2. Який зв'язок існує між IIoT та КФВС?
3. Які пристрої можуть входити до промислової IIoT-системи?
4. Які функції виконує IIoT-шлюз?
5. Чим ідентифікатор виробничого активу відрізняється від його мережевої адреси?
6. Що розуміють під edge computing?
7. Які задачі доцільно виконувати на edge-рівні?
8. Чому передавання всіх первинних сенсорних даних у хмару не завжди раціональне?
9. Чим fog computing відрізняється від edge computing?
10. Для яких задач доцільно використовувати cloud computing?
11. Які функції необхідно залишати на ПЛК?
12. Від яких складових залежить повний час виконання віддаленої задачі?
13. Як периферійне оброблення може зменшити мережевий трафік?
14. Які обмеження мають edge-пристрої?
15. За якими критеріями визначається місце виконання алгоритму?
16. Чому для КФВС доцільне спільне використання PLC–Edge–Fog–Cloud архітектури?

Список використаних джерел

1. Pivoto D. G. S., de Almeida L. F. F., Righi R. R., Rodrigues J. J. P. C., Lugli A. B., Alberti A. M. Cyber-physical systems architectures for industrial internet of things applications in Industry 4.0: A literature review. *Journal of Manufacturing Systems*. 2021. Vol. 58. P. 176–192. DOI: 10.1016/j.jmsy.2020.11.017.
2. Javaid M., Haleem A., Singh R. P., Rab S., Suman R. Upgrading the manufacturing sector via applications of Industrial Internet of Things (IIoT). *Sensors International*. 2021. Vol. 2. Article 100129. DOI: 10.1016/j.sintl.2021.100129.
3. Nain G., Pattanaik K. K., Sharma G. K. Towards edge computing in intelligent manufacturing: Past, present and future. *Journal of Manufacturing Systems*. 2022. Vol. 62. P. 588–611. DOI: 10.1016/j.jmsy.2022.01.010.
4. Kubiak K., Dec G., Stadnicka D. Possible Applications of Edge Computing in the Manufacturing Industry—Systematic Literature Review. *Sensors*. 2022. Vol. 22, No. 7. Article 2445. DOI: 10.3390/s22072445.
5. Caiza G., Saeteros M., Oñate W., Garcia M. V. Fog computing at industrial level, architecture, latency, energy, and security: A review. *Heliyon*. 2020. Vol. 6, No. 4. Article e03706. DOI: 10.1016/j.heliyon.2020.e03706.
6. Kumar R., Agrawal N. Analysis of multi-dimensional Industrial IoT (IIoT) data in Edge–Fog–Cloud based architectural frameworks: A survey on

current state and research challenges. *Journal of Industrial Information Integration*. 2023. Vol. 35. Article 100504. DOI: 10.1016/j.jii.2023.100504.

7. Alotaibi B. A Survey on Industrial Internet of Things Security: Requirements, Attacks, AI-Based Solutions, and Edge Computing Opportunities. *Sensors*. 2023. Vol. 23, No. 17. Article 7470. DOI: 10.3390/s23177470.

8. Raja Sree T. Role of fog-assisted Industrial Internet of Things: A systematic review. *Transactions on Emerging Telecommunications Technologies*. 2022. DOI: 10.1002/ett.4611

ЛЕКЦІЯ 9. ІНТЕГРАЦІЯ ВИРОБНИЧИХ СИСТЕМ: SCADA, MES, ERP ТА ІНФОРМАЦІЙНІ МОДЕЛІ

Мета лекції

Мета лекції полягає у формуванні уявлення про принципи інформаційної інтеграції кіберфізичних виробничих систем, функції SCADA, MES та ERP і характер обміну між ними. Розглядаються вертикальна та горизонтальна інтеграція, модель ISA-95/IEC 62264, інформаційні моделі виробничих об'єктів, семантична сумісність і способи передавання технологічних, виробничих та корпоративних даних.

У попередніх лекціях окремо розглядалися польове обладнання, ПЛК, промислові мережі та ІоТ. Проте сама наявність зв'язку між компонентами ще не створює інтегрованого виробництва. Дані повинні переходити між різними рівнями без втрати фізичного змісту, а системи — однозначно трактувати виробничі замовлення, ресурси, стани обладнання та результати виконання операцій. Саме ця проблема лежить в основі інтеграції SCADA–MES–ERP [1–3].

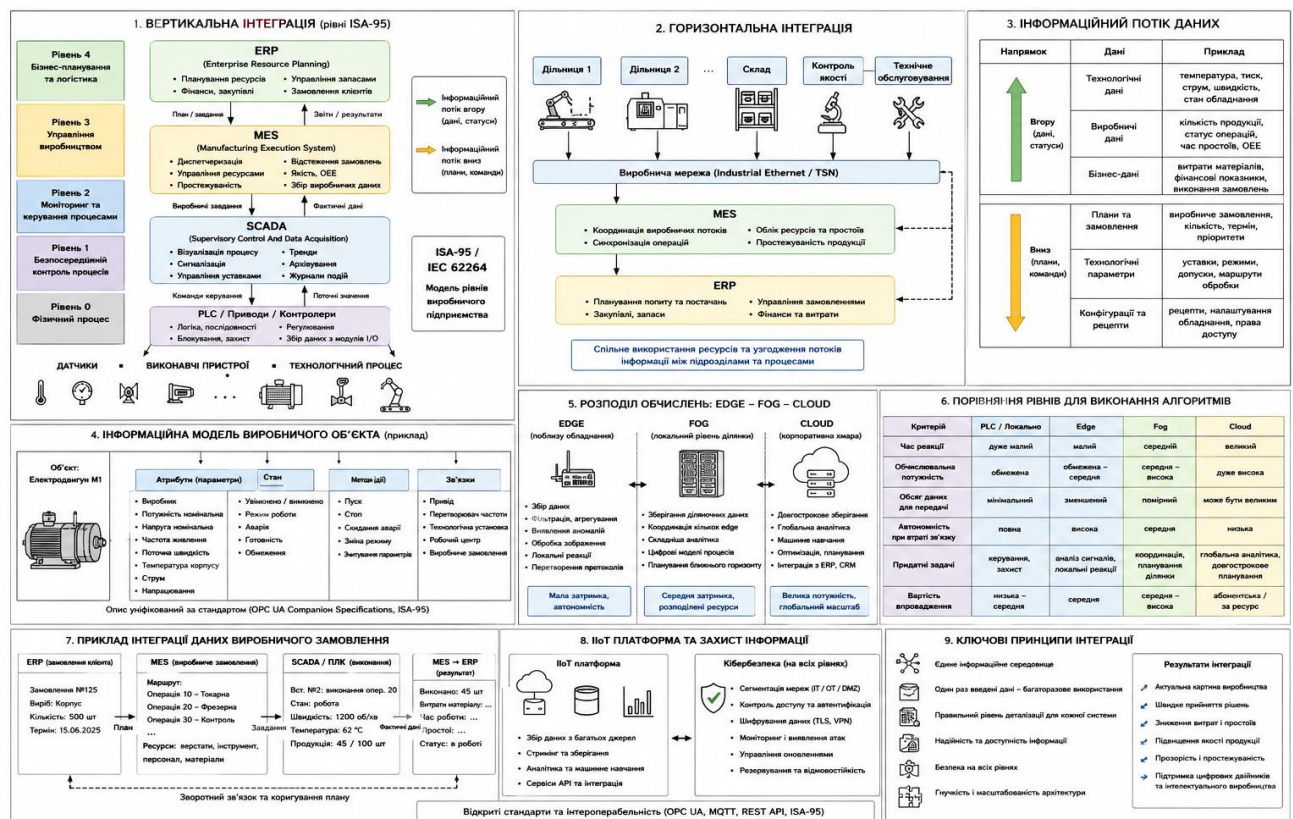


Рисунок 9.1 – Інтеграція виробничих систем: scada, mes, erp та інформаційні моделі

9.1. Вертикальна та горизонтальна інтеграція виробництва

Інтеграція виробничих систем має два основних напрямки — вертикальний і горизонтальний.

Вертикальна інтеграція поєднує рівні одного підприємства: від фізичного обладнання і локального керування до виробничого планування та корпоративного управління. У спрощеному вигляді:

датчики та виконавчі пристрої → ПЛК → SCADA → MES → ERP.

При цьому напрямок передавання інформації не є лише висхідним. Верхні системи передають вниз виробничі завдання, рецепти, пріоритети та планові показники, а нижні — фактичний стан обладнання, кількість виготовленої продукції, параметри процесу та інформацію про відмови.

Вертикальна інтеграція потрібна для того, щоб рішення верхнього рівня спиралося не на застарілу або вручну введену інформацію, а на фактичний стан виробництва. Дослідження smart manufacturing розглядають саме зв'язок від рівня обладнання до ERP як одну з характерних ознак вертикально інтегрованого виробництва [1].

Повний час проходження інформації між рівнями:

$$T_{\text{інт}} = T_{\text{зб}} + T_{\text{перед}} + T_{\text{перет}} + T_{\text{зап}},$$

де $T_{\text{інт}}$ — час інтеграційного циклу; $T_{\text{зб}}$ — час отримання даних від джерела; $T_{\text{перед}}$ — час передавання між системами; $T_{\text{перет}}$ — час перетворення формату або інформаційної моделі; $T_{\text{зап}}$ — час запису та актуалізації даних у системі-одержувачі.

Для виробничого планування цей час може вимірюватися секундами або хвилинами. Для локального керування така архітектура неприйнятна — швидкі регулятори повинні залишатися на нижньому рівні.

Горизонтальна інтеграція поєднує системи одного функціонального рівня або сусідні виробничі ланки. Це може бути взаємодія декількох виробничих дільниць, складу, контролю якості, технічного обслуговування та внутрішньої логістики.

У ширшому розумінні горизонтальна інтеграція може охоплювати постачальників, виробника, логістичні організації та споживачів. Тоді інформаційний потік супроводжує матеріальний потік уздовж ланцюга створення продукції [1, 4].

Для КФВС вертикальна і горизонтальна інтеграція доповнюють одна одну. Перша забезпечує зв'язок між рівнями керування, друга — між виробничими функціями та об'єктами.

9.2. SCADA-системи та диспетчерське керування

SCADA — Supervisory Control and Data Acquisition — виконує диспетчерське спостереження та керування технологічним процесом. Вона знаходиться над локальними ПЛК і отримує інформацію вже після того, як польові сигнали були виміряні, оцифровані та первинно оброблені.

Типові функції SCADA:

- відображення технологічного процесу;
- збір поточних параметрів;
- ведення трендів;
- сигналізація аварійних станів;
- архівування;
- формування команд оператором;
- керування рецептами;
- ведення журналу подій.

SCADA не повинна підміняти локальний контролер. Наприклад, підтримання тиску або швидкості двигуна здійснюється ПЛК чи приводом, тоді як SCADA задає уставку, показує результат і фіксує відхилення.

Таке розділення визначається різними часовими вимогами. При короткочасній втраті SCADA локальний ПЛК повинен продовжувати виконання безпечного технологічного алгоритму.

Для диспетчерської системи важлива актуальність інформації. Її можна характеризувати інтервалом:

$$\Delta t_{\text{акт}} = t_{\text{пот}} - t_{\text{дж}},$$

де $\Delta t_{\text{акт}}$ — вік отриманих даних; $t_{\text{пот}}$ — поточний момент використання інформації; $t_{\text{дж}}$ — час формування значення у джерелі.

Якщо SCADA показує значення, отримане значно раніше, оператор фактично спостерігає не поточний, а минулий стан технологічного процесу.

Через це дані SCADA повинні містити часові мітки, а архітектура — розрізняти момент вимірювання і момент відображення.

У системі верхнього рівня SCADA є насамперед джерелом технологічної інформації. MES використовує вже не окремий сигнал датчика сам по собі, а його виробничий контекст: до якого замовлення належала операція, який виріб оброблявся і який ресурс використовувався.

9.3. MES як рівень оперативного управління виробництвом

Manufacturing Execution System — MES — займає проміжне положення між технологічним керуванням і корпоративним плануванням.

Основна задача MES полягає у перетворенні виробничого плану на конкретні операції та зборі фактичних результатів їх виконання.

До функцій MES можуть входити:

- детальне планування операцій;
- диспетчеризація виробничих замовлень;
- управління ресурсами;
- відстеження проходження продукції;
- збір фактичних виробничих даних;
- контроль якості;
- облік простоїв;
- простежуваність матеріалів;
- аналіз виробничої ефективності.

У сучасних роботах MES розглядається не як ізольована монолітна програма, а як інтеграційний рівень між ERP і технологічними системами, причому для гнучкого виробництва досліджуються модульні та сервісно-орієнтовані реалізації [2].

MES працює з іншою семантикою даних, ніж SCADA.

SCADA може повідомляти:

«двигун M2 працює, струм 12,4 А».

MES повинна пов'язати цей факт із виробництвом:

«операція № 40 замовлення № 125 виконується на робочому центрі C2».

Отже, при переході від SCADA до MES фізичні сигнали набувають виробничого контексту.

Одним із показників, який може розраховуватися на основі MES-даних, є загальна ефективність обладнання:

$$OEE = A_d P_{пр} Q_{як},$$

де OEE — Overall Equipment Effectiveness; A_d — доступність обладнання; $P_{пр}$ — продуктивність відносно нормативної; $Q_{як}$ — частка придатної продукції.

Ця формула показує, чому одного сигналу «машина увімкнена» недостатньо для оцінювання виробництва. Потрібні відомості про фактичний час роботи, темп виконання операцій та якість продукції.

9.4. Взаємодія MES і ERP

Enterprise Resource Planning — ERP — працює на рівні бізнес-планування і логістики. Вона оперує замовленнями, матеріалами, запасами, фінансовими та кадровими ресурсами, планами виробництва й поставок.

ERP відповідає переважно на питання:

що, скільки і коли потрібно виготовити?

MES вирішує інше:

як конкретно виконати це завдання доступними виробничими ресурсами?

Тому типовий обмін має двосторонню структуру.

ERP → MES:

- виробниче замовлення;
- номенклатура;
- кількість;
- термін;
- специфікація;
- плановий маршрут.

MES → ERP:

- фактично виготовлена кількість;
- фактичні витрати матеріалів;
- час виконання;
- простої;
- статус замовлення;
- фактична виробнича продуктивність.

Стандарт ISA-95 приділяє значну увагу саме межі між manufacturing operations management і business planning and logistics, тобто рівнями, на яких у типовій архітектурі знаходяться MES та ERP [3]. У 2025 році ISA

опублікувала оновлену редакцію Part 1 — ANSI/ISA-95.00.01-2025, що актуалізує моделі й термінологію інтеграції enterprise та control systems.

Ступінь виконання виробничого замовлення можна характеризувати:

$$K_{\text{вик}} = \frac{N_{\text{факт}}}{N_{\text{план}}},$$

де $K_{\text{вик}}$ — коефіцієнт виконання замовлення; $N_{\text{факт}}$ — фактична кількість виготовлених одиниць; $N_{\text{план}}$ — планова кількість.

Для ERP цей показник може бути достатнім для оцінки замовлення. MES, однак, повинна знати значно більше: які операції вже виконані, на яких машинах, які матеріали використані та які причини відхилення від плану.

Саме тому пряме підключення ERP до тисяч польових сигналів зазвичай недоцільне. MES виконує функцію переходу від технологічної деталізації до виробничої інформації.

9.5. ISA-95 та структуризація рівнів виробничого управління

ISA-95, міжнародним відповідником якої є серія IEC 62264, створена для формалізації інтеграції виробничого керування та корпоративних функцій.

Модель виділяє рівні 0–4:

рівень 0 — фізичний виробничий процес;

рівень 1 — вимірювання та безпосередній вплив на процес;

рівень 2 — моніторинг і автоматичне керування;

рівень 3 — Manufacturing Operations Management;

рівень 4 — Business Planning and Logistics.

ISA прямо зазначає, що модель орієнтується насамперед на види діяльності, а не на конкретні технології. Через це її можна застосовувати і до сучасних розподілених систем, хоча фізична архітектура вже не обов'язково нагадує жорстку піраміду [3].

Цей момент принциповий. Неправильно трактувати ISA-95 так, ніби кожний інформаційний пакет обов'язково повинен послідовно пройти всі рівні.

Наприклад, edge-вузол може напряму передавати діагностичну інформацію у корпоративну систему. Проте модель ISA-95 все одно дозволяє визначити, до якої функціональної області належить ця інформація.

Серія стандартів описує не лише рівні, а й об'єкти інформаційного обміну, виробничі операції, транзакції та сервіси.

IEC 62264-6:2020 визначає технологічно незалежну модель сервісів обміну повідомленнями для взаємодії між застосунками manufacturing operations domain і системами інших областей [4].

Отже, ISA-95/IEC 62264 доцільно розглядати не як схему кабельних з'єднань, а як мову опису функціональної та інформаційної інтеграції підприємства.

9.6. Інформаційні моделі технологічних об'єктів

Для інтеграції недостатньо передавати числові значення. Потрібно однаково описати їх зміст.

Інформаційну модель можна формально представити:

$$\mathcal{M}_I = (\mathcal{O}, \mathcal{A}, \mathcal{R}_I),$$

де $\mathcal{M}_I$ — інформаційна модель; $\mathcal{O}$ — множина об'єктів; $\mathcal{A}$ — множина їх атрибутів; $\mathcal{R}_I$ — множина семантичних зв'язків між об'єктами.

Наприклад, об'єкт «Електродвигун М1» може мати атрибути:

- виробник;
- номінальна потужність;
- поточна швидкість;
- температура;
- стан;
- виробнича ділянка.

Він також може бути пов'язаний із приводом, технологічною установкою, операцією та виробничим замовленням.

Такий опис значно відрізняється від звичайної таблиці тегів.

OPC UA надає механізм представлення об'єктів, змінних, методів, типів та зв'язків між ними. На цій основі створюються галузеві Companion Specifications, що задають уніфіковані інформаційні моделі конкретних класів обладнання.

Практичне дослідження інтеграції CNC-верстатів показало, що інформаційна модель OPC UA дозволяє уніфікувати моніторинг обладнання різних типів і знизити залежність верхнього програмного рівня від пропрієтарних інтерфейсів конкретних машин [5].

Отже, інформаційна модель виконує функцію посередника між фізичною структурою виробництва та програмними системами.

9.7. Семантична сумісність компонентів

Інтероперабельність має декілька рівнів.

Технічна сумісність означає, що пристрої фізично можуть передавати дані.

Синтаксична сумісність означає, що вони однаково розуміють формат повідомлення.

Семантична сумісність означає, що обидві сторони однаково трактують зміст інформації.

Розглянемо два параметри:

Speed = 1500

і

RotationalSpeed = 1500 rpm.

На рівні людини можна здогадатися, що вони можуть позначати одну величину. Для автоматичної інтеграції цього недостатньо.

Потрібне формальне відображення:

$$m: \mathcal{O}_A \rightarrow \mathcal{O}_B,$$

де m — функція семантичного відображення; $\mathcal{O}_A$ і $\mathcal{O}_B$ — множини об'єктів двох інформаційних моделей.

Якщо кожна пара систем потребує власного ручного відображення, складність інтеграції швидко зростає. Саме тому застосовуються стандартизовані інформаційні моделі та спільна термінологія.

OPC UA Companion Specifications є одним із практичних способів такого узгодження. Наприклад, сучасні специфікації OPC Foundation визначають інформаційні моделі для машин, адитивного виробництва, різального інструменту та інших класів виробничих активів.

Семантична сумісність особливо потрібна цифровим двійникам та системам машинного навчання. Якщо одна система трактує параметр як температуру корпусу, а інша — як температуру обмотки, формально правильний обмін числами створить фізично неправильний набір даних.

9.8. Обмін технологічними, виробничими та корпоративними даними

У повністю інтегрованих КФВС одночасно існують щонайменше три класи даних.

Технологічні дані

Вони безпосередньо описують фізичний процес:

- температура;
- тиск;
- швидкість;
- струм;
- стан датчика;
- положення механізму.

Такі дані надходять переважно від ПЛК, приводів і SCADA.

Виробничі дані

Вони описують виконання операцій:

- номер замовлення;
- виріб;
- партія;
- технологічний маршрут;
- стан операції;
- використаний ресурс;
- кількість придатних виробів;
- тривалість простою.

Це основна інформаційна область MES.

Корпоративні дані

До них належать:

- замовлення клієнтів;
- плани;
- матеріальні ресурси;
- запаси;
- закупівлі;
- фінансові показники;
- логістичні дані.

З ними переважно працює ERP.

При інтеграції потрібно не механічно копіювати всі дані знизу вгору, а перетворювати їх рівень деталізації.

Наприклад:

SCADA: «датчик виробив сигнал о 10:15:32»;

MES: «операція завершена о 10:15:32»;

ERP: «виготовлено одну одиницю замовленої продукції».

Кожний наступний рівень використовує більш агрегований зміст.

Коефіцієнт інформаційного агрегування:

$$K_{\text{агр}} = \frac{V_{\text{верх}}}{V_{\text{низ}}},$$

де $K_{\text{агр}}$ — відношення обсягу даних після агрегування до початкового;
 $V_{\text{верх}}$ — обсяг інформації, переданої на верхній рівень; $V_{\text{низ}}$ — обсяг первинних даних нижнього рівня.

Зазвичай $K_{\text{агр}}$: тисячі сенсорних вимірювань можуть бути перетворені на один виробничий показник.

Проте зворотний потік має інший характер. ERP може передати одне замовлення, яке MES розкладає на десятки технологічних операцій, а ПЛК — на сотні конкретних керувальних дій.

Тому інтеграція не є простим рухом інформації «вгору». Це послідовне перетворення змісту між різними рівнями абстракції.

Дослідження сучасних MES показують тенденцію до сервісно-орієнтованої інтеграції, де окремі виробничі функції взаємодіють через стандартизовані інтерфейси, а OPC UA може використовуватися для наскрізного обміну між технологічними й виробничими компонентами [2].

При цьому не кожна система повинна мати прямий доступ до кожної іншої. Архітектура має визначати власника даних, джерело істини, допустимі напрямки запису та поведінку при втраті зв'язку.

Якщо одна й та сама технологічна величина незалежно редагується в SCADA, MES і ERP, неминуче виникає проблема узгодженості. Тому для кожного параметра повинно бути визначено, яка система є його первинним джерелом.

Висновки до лекції 9

Інтеграція кіберфізичного виробництва полягає не лише у створенні мережесх зв'язків, а в узгодженні функцій та інформаційних моделей різних систем. Вертикальна інтеграція поєднує обладнання, локальне керування, SCADA, MES і ERP, тоді як горизонтальна забезпечує взаємодію виробничих підрозділів та ланок технологічного і логістичного процесу.

SCADA працює переважно з фактичним станом технологічного процесу. MES додає до цих даних виробничий контекст і пов'язує фізичні операції із замовленнями та ресурсами. ERP працює на рівні планування, логістики та корпоративного управління. Відмінність між цими системами визначається не програмним продуктом як таким, а характером функцій і даних.

ISA-95/IEC 62264 формує технологічно незалежну основу для опису меж між виробничим керуванням та бізнес-функціями. Її рівні не потрібно трактувати як жорстку фізичну піраміду; вони насамперед задають функціональні області та зміст інформаційного обміну.

Інформаційні моделі дозволяють перейти від передавання окремих чисел до формального цифрового опису виробничого активу. Семантична сумісність забезпечує однакове трактування даних різними системами й стає необхідною умовою масштабованої інтеграції.

У повній КФВС технологічні дані поступово перетворюються на виробничі, а виробничі — на корпоративну інформацію. Зворотний процес перетворює виробничі плани на конкретні технологічні дії. Саме

двосторонність цього зв'язку формує інформаційно інтегроване виробництво.

Контрольні питання

1. Що розуміють під вертикальною інтеграцією виробничої системи?
2. У чому полягає горизонтальна інтеграція?
3. Які функції виконує SCADA?
4. Чому швидкі контури регулювання не слід переносити до SCADA?
5. Яке функціональне призначення MES?
6. Чим дані MES відрізняються від технологічних сигналів SCADA?
7. Яку роль виконує ERP?
8. Яка інформація передається від ERP до MES?
9. Які дані MES повертає до ERP?
10. Які рівні виділяє ISA-95?
11. Яке місце в ISA-95 займає Manufacturing Operations Management?
12. Чому ISA-95 не слід трактувати лише як фізичну піраміду обладнання?
13. Що являє собою інформаційна модель виробничого об'єкта?
14. Чим технічна сумісність відрізняється від семантичної?
15. Для чого використовуються OPC UA Companion Specifications?
16. Чим відрізняються технологічні, виробничі та корпоративні дані?
17. Чому при інтеграції необхідно визначати первинне джерело кожного параметра?

Список використаних джерел

1. Valero M. R., Newman S. T., Nassehi A. Link4Smart: A New Framework for Smart Manufacturing Linking Industry 4.0 Relevant Technologies. *Procedia CIRP*. 2022. Vol. 107. P. 1594–1599. DOI: 10.1016/j.procir.2022.05.196.
2. Chohan B. S., Xu X., Lu Y. MES Dynamic interoperability for SMEs in the Factory of the Future perspective. *Procedia CIRP*. 2022. Vol. 107. P. 1329–1335. DOI: 10.1016/j.procir.2022.05.153.
3. ANSI/ISA-95.00.01-2025 (IEC 62264-1 Modified). *Enterprise-Control System Integration — Part 1: Models and Terminology*. Research Triangle Park : International Society of Automation, 2025.
4. IEC 62264-6:2020. *Enterprise-control system integration — Part 6: Messaging service model*. Geneva : International Electrotechnical Commission, 2020. 50 p.
5. Martins A., Lucas J., Costelha H., Neves C. CNC Machines Integration in Smart Factories using OPC UA. *Journal of Industrial Information Integration*. 2023. Vol. 34. Article 100482. DOI: 10.1016/j.jii.2023.100482.
6. Dillinger F., Bernhard O., Kagerer M. et al. Industry 4.0 implementation sequence for manufacturing companies. *Production Engineering*. 2022. Vol. 16. P. 705–718.

7. Alqoud A., Schaefer D., Milisavljevic-Syed J. Industry 4.0: A Review of Digital Retrofitting Solutions for Legacy Manufacturing Systems. *Advances in Transdisciplinary Engineering*. 2022. DOI: 10.3233/ATDE220557.

8. *OPC UA for Additive Manufacturing. OPC 40540*. OPC Foundation, 2024.

ЛЕКЦІЯ 10. ЦИФРОВІ ДВІЙНИКИ КІБЕРФІЗИЧНИХ ВИРОБНИЧИХ СИСТЕМ

Мета лекції

Мета лекції полягає у формуванні уявлення про цифровий двійник як засіб узгодженого представлення фізичного виробничого об'єкта, його математичних та інформаційних моделей і даних, що надходять під час експлуатації. Розглядаються відмінності між цифровою моделлю, цифровою тінню та цифровим двійником, принципи синхронізації фізичного і цифрового станів, способи оцінювання та прогнозування стану, віртуальне налагодження систем керування і межі достовірності цифрового представлення.

Цифровий двійник не слід ототожнювати з будь-якою тривимірною моделлю обладнання або набором даних у SCADA. Його практичний зміст виникає тоді, коли цифрове представлення пов'язане з реальним об'єктом даними, оновлюється відповідно до фактичного стану та використовується для аналізу, прогнозування або формування рішень [1–3]. Огляд сучасних досліджень показує, що термінологія цифрових двійників усе ще не повністю уніфікована, тому при проєктуванні необхідно явно визначати, які саме інформаційні та функціональні зв'язки реалізовані [3].

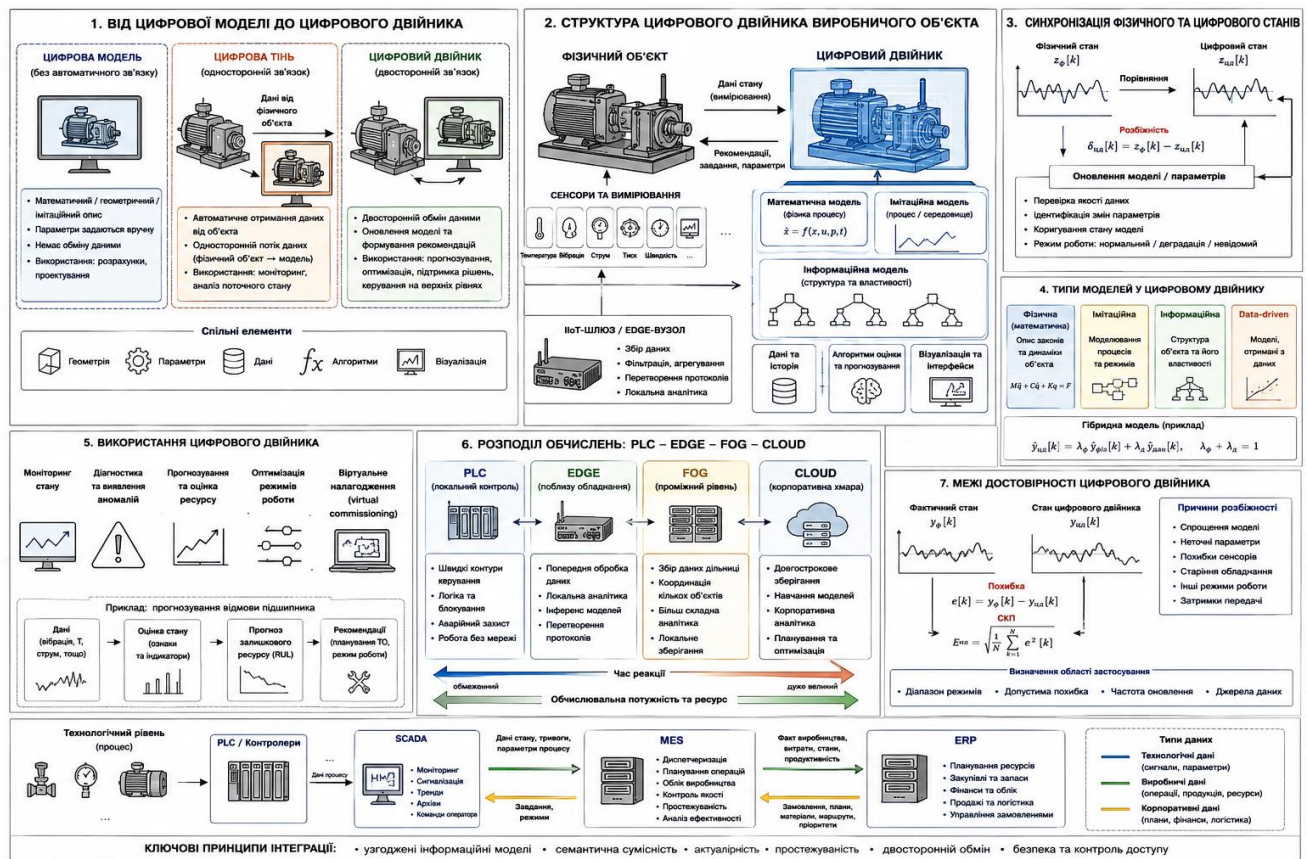


Рисунок 10.1 – Цифрові двійники кіберфізичних виробничих систем

10.1. Поняття цифрової моделі, цифрової тіні та цифрового двійника

Цифрове представлення фізичного виробничого об'єкта може мати різний рівень зв'язку з реальним обладнанням. Для навчального курсу доцільно розрізнити три поняття: цифрова модель, цифрова тінь і цифровий двійник. Такий поділ потрібно сприймати як робочу інженерну класифікацію, оскільки в літературі межі між цими поняттями трактуються не повністю однаково [1, 3].

Цифрова модель є математичним, геометричним, логічним або імітаційним описом фізичного об'єкта, який не обов'язково має автоматичний зв'язок із реальним обладнанням.

Наприклад, математична модель електропривода може використовуватися для розрахунку перехідного процесу. Її параметри задає інженер, а зміни реального стану двигуна не потрапляють у модель автоматично.

Цифрова тінь передбачає автоматизоване надходження інформації від фізичного об'єкта до цифрового представлення. Напрямок зв'язку переважно має вигляд:

фізичний об'єкт → цифрове представлення.

Наприклад, модель верстата може регулярно оновлювати інформацію про швидкість шпинделя, температуру, навантаження та кількість виконаних операцій.

Цифровий двійник передбачає більш тісний двосторонній зв'язок, коли цифрова система не лише отримує дані про реальний стан, але й використовується для формування рекомендацій, прогнозів, оптимізаційних рішень або керувальних завдань:

фізичний об'єкт ↔ цифровий двійник.

ISO 23247-1:2021 визначає загальні принципи й вимоги до framework цифрових двійників для виробництва та розглядає цифровий двійник саме в контексті виробничих об'єктів і систем [4].

При цьому двосторонній зв'язок не означає, що цифровий двійник повинен безпосередньо керувати фізичним обладнанням. У багатьох системах він формує рекомендацію, а остаточне рішення виконується ПЛК, MES або оператором.

10.2. Структура цифрового двійника виробничого об'єкта

Цифровий двійник доцільно розглядати не як одну програму, а як систему взаємопов'язаних компонентів.

До його складу можуть входити:

- фізичний виробничий об'єкт;
- сенсорна система;
- математична модель;
- інформаційна модель;
- історичні дані;
- алгоритми оцінювання стану;
- прогнозні моделі;
- засоби візуалізації;
- інтерфейси з ПЛК, SCADA та MES;
- механізми синхронізації.

Огляди цифрових двійників у виробництві показують, що їх структура залежить від задачі та етапу життєвого циклу об'єкта. Один двійник може бути орієнтований на проєктування, інший — на виробниче керування, діагностування або технічне обслуговування [1, 2].

Стан цифрового представлення:

$$\mathbf{z}_{\text{цд}}[k] = \mathbf{F}_{\text{цд}}(\mathbf{z}_{\text{цд}}[k-1], \mathbf{y}[k], \mathbf{p}_{\text{цд}}),$$

де $\mathbf{z}_{\text{цд}}[k]$ — внутрішній стан цифрового двійника; $\mathbf{F}_{\text{цд}}$ — функція його оновлення; $\mathbf{p}_{\text{цд}}$ — вектор параметрів цифрового двійника.

У цій структурі нові дані від сенсорів не просто записуються в базу, а змінюють цифровий стан. Наприклад, може оновлюватися оцінка температурного режиму, навантаження, рівня зношування або доступності виробничого ресурсу.

При проєктуванні необхідно визначити, які властивості фізичного об'єкта справді потрібні цифровому двійнику. Повна геометрична модель верстата не обов'язкова для задачі прогнозування температури підшипника. Навпаки, для аналізу зіткнень робота геометричні характеристики мають принципове значення.

Тому структура двійника визначається не максимально можливою кількістю даних, а поставленою інженерною задачею.

10.3. Синхронізація фізичного об'єкта та його цифрової моделі

Ключова властивість цифрового двійника — відповідність цифрового стану фактичному стану фізичного об'єкта.

Синхронізація передбачає:

- отримання актуальних сенсорних даних;
- часову прив'язку;
- перевірку якості вимірювань;
- оновлення параметрів;
- визначення режиму роботи;
- коригування внутрішнього стану моделі.

Нехай цифровий двійник і фізичний об'єкт мають відповідні порівнювані параметри. Їх розбіжність:

$$\delta_{\text{цд}}[k] = \mathbf{z}_{\text{ф}}[k] - \mathbf{z}_{\text{цд}}[k],$$

де $\delta_{\text{цд}}[k]$ — вектор розбіжності фізичного та цифрового станів; $\mathbf{z}_{\text{ф}}[k]$ — доступне цифрове представлення фактичного стану фізичного об'єкта.

Якщо розбіжність систематично зростає, можливі декілька причин:

- змінилися параметри фізичного об'єкта;
- модель недостатньо точна;
- сенсор формує некоректні дані;
- виникла несправність;
- цифровий двійник працює з неактуальною конфігурацією.

Отже, відхилення між моделлю та фізичним об'єктом не завжди означає несправність обладнання. Спочатку необхідно перевірити саму модель і вимірювальний тракт.

Для оцінювання ступеня узгодження можна використовувати норму розбіжності:

$$D_{\text{цд}}[k] = \|\delta_{\text{цд}}[k]\|,$$

де $D_{\text{цд}}[k]$ — числова міра розходження цифрового та фізичного станів.

Граничне допустиме значення цього показника повинно визначатися відповідно до конкретної задачі. Для грубого виробничого планування допустиме відхилення може бути значно більшим, ніж для високоточного прогнозування руху.

Цифровий двійник також повинен знати часову актуальність даних. Якщо мережевий зв'язок втрачено, цифрове представлення не зупиняється фізично разом з об'єктом. Тому модель повинна відрізнати вимірний стан від стану, прогнозованого після останнього доступного вимірювання.

10.4. Математичні, імітаційні та інформаційні моделі цифрового двійника

Цифровий двійник може використовувати декілька типів моделей одночасно.

Математична модель описує фізичні закономірності та динаміку об'єкта.

Імітаційна модель відтворює поведінку системи або виробничого процесу в заданих сценаріях.

Інформаційна модель описує структуру активу, його властивості, параметри та зв'язки з іншими виробничими об'єктами.

Модель, отримана з даних, формує залежності на основі накопичених вимірювань.

На практиці ці способи доцільно комбінувати. Наприклад, фізична модель може описувати механіку електропривода, інформаційна — його положення у виробничій структурі, а статистична — залежність температури підшипника від навантаження.

У цифровому двійнику може застосовуватися гібридна модель:

$$\hat{y}_{\text{цд}}[k] = \lambda_{\text{ф}} \hat{y}_{\text{фіз}}[k] + \lambda_{\text{д}} \hat{y}_{\text{дан}}[k],$$

де $\hat{y}_{\text{цд}}[k]$ — прогноз цифрового двійника; $\hat{y}_{\text{фіз}}[k]$ — результат фізичної моделі; $\hat{y}_{\text{дан}}[k]$ — результат моделі, отриманої з даних; $\lambda_{\text{ф}}$ і $\lambda_{\text{д}}$ — вагові коефіцієнти відповідних складових.

Для нормованої комбінації:

$$\lambda_{\text{ф}} + \lambda_{\text{д}} = 1.$$

Поєднання фізичних і data-driven моделей дозволяє використати відомі закономірності процесу та водночас коригувати результат за експлуатаційними даними.

Цифровий двійник не повинен бути складнішим, ніж потрібно для його функції. Огляд Leng та співавторів показує, що цифрові двійники можуть охоплювати функціональні, структурні, поведінкові та керувальні аспекти виробничої системи, але необхідний рівень деталізації залежить від конкретного застосування [2].

10.5. Оцінювання стану об'єкта за вимірювальними даними

Не всі параметри фізичного об'єкта можна виміряти безпосередньо. Частина з них оцінюється за математичною моделлю та доступними сенсорними сигналами.

У цифровому двійнику оцінювання стану виконує дві задачі:
відновлює недоступні для прямого вимірювання параметри;
зменшує вплив шумів та окремих некоректних вимірювань.

Поточну оцінку можна розглядати як результат поєднання прогнозу моделі та нового вимірювання:

$$\hat{\mathbf{x}}_{\text{кор}}[k] = \hat{\mathbf{x}}_{\text{пр}}[k] + \mathbf{K}_{\text{оц}} (\mathbf{y}[k] - \hat{\mathbf{y}}_{\text{пр}}[k]),$$

де $\hat{\mathbf{x}}_{\text{кор}}[k]$ — скоригована оцінка стану; $\hat{\mathbf{x}}_{\text{пр}}[k]$ — прогнозована оцінка до отримання нового вимірювання; $\mathbf{K}_{\text{оц}}$ — матриця коригування; $\hat{\mathbf{y}}_{\text{пр}}[k]$ — прогноз вимірюваного виходу.

Це співвідношення показує загальну ідею спостерігача: цифрова модель прогнозує стан, а нове вимірювання коригує прогноз.

Якщо модель і сенсорні дані узгоджуються, корекція невелика. Якщо між ними виникає значна різниця, цифровий стан змінюється сильніше або система формує діагностичну ознаку.

Саме тому цифровий двійник потребує достовірної сенсорної системи. Збільшення кількості вимірювань не обов'язково покращує оцінювання, якщо вони дублюють одну й ту саму інформацію або мають низьку якість.

10.6. Використання цифрових двійників для прогнозування поведінки системи

Одна з найбільш цінних властивостей цифрового двійника — можливість оцінювати не тільки поточний, а й майбутній стан.

Прогноз може стосуватися:

- температури;
- навантаження;
- споживання енергії;
- виробничої продуктивності;

- накопичення зношування;
- часу завершення операції;
- імовірності переходу в небажаний режим.

Прогноз на $N_{\text{пр}} H_{\text{пр}}$ дискретних кроків можна записати:

$$\hat{\mathbf{x}}[k + H_{\text{пр}}] = \mathbf{F}_{\text{пр}}^{H_{\text{пр}}} \left(\hat{\mathbf{x}}[k], \mathbf{u}[k : k + H_{\text{пр}} - 1] \right),$$

де $H_{\text{пр}}$ — горизонт прогнозування; $\mathbf{F}_{\text{пр}}^{H_{\text{пр}}}$ — оператор багатокрокового прогнозу.

Із збільшенням горизонту зазвичай зростає невизначеність результату. Причина полягає у накопиченні похибки моделі та неможливості точно передбачити майбутні збурення.

Тому прогноз цифрового двійника потрібно супроводжувати інформацією про його достовірність.

Наприклад, твердження «температура через годину становитиме 85 °C» є надто категоричним, якщо невизначеність моделі значна. Для практичного використання краще оцінювати допустимий діапазон або ймовірність перевищення критичного рівня.

Огляди цифрових двійників відносять прогнозування, моніторинг та оптимізацію до основних функціональних напрямів їх застосування [1, 3].

10.7. Віртуальне налагодження алгоритмів керування

Віртуальне налагодження — virtual commissioning — дозволяє перевіряти алгоритми керування до повного введення реального обладнання в експлуатацію.

Замість фізичного об'єкта до ПЛК або програмної системи підключається його цифрова модель.

Типова послідовність має вигляд:

модель об'єкта ↔ ПЛК або емулятор контролера ↔ алгоритм керування.

Це дозволяє перевірити:

- логіку технологічної послідовності;
- міжблокування;
- переходи між режимами;
- реакцію на аварійні сигнали;
- взаємодію декількох механізмів;
- правильність сигналів введення-виведення.

Віртуальне налагодження зменшує залежність тестування програмної частини від готовності фізичного обладнання. Дослідження Barbieri та

співавторів описує послідовну методологію інтеграції цифрового двійника у виробничу систему саме через virtual commissioning [5].

Інші роботи демонструють застосування цифрових двійників для дистанційного напівфізичного налагодження виробничих систем [6].

Якість віртуального тестування залежить від адекватності моделі. Якщо модель не враховує реальну затримку датчика, люфт, обмеження привода або мережеве запізнення, контролер може успішно пройти віртуальний тест і працювати інакше на реальному об'єкті.

Тому результати virtual commissioning не скасовують фізичні пусконалагоджувальні випробування. Вони дозволяють перенести значну частину перевірок на більш ранній етап.

10.8. Межі достовірності цифрового двійника

Цифровий двійник не є абсолютно точним цифровим еквівалентом фізичного об'єкта. Він завжди має обмежену область достовірності.

Причинами розбіжності можуть бути:

спрощення математичної моделі;

похибки сенсорів;

неточність параметрів;

старіння обладнання;

зміна навантаження;

неповні дані;

затримки синхронізації;

режими, відсутні під час валідації.

Середньоквадратичну похибку цифрового двійника можна визначати аналогічно оцінюванню інших моделей:

$$E_{\text{цд}} = \sqrt{\frac{1}{N_{\text{цд}}} \sum_{k=1}^{N_{\text{цд}}} (y_{\text{ф}}[k] - y_{\text{цд}}[k])^2},$$

де $E_{\text{цд}}$ — середньоквадратична похибка цифрового двійника; $N_{\text{цд}}$ — кількість перевірочних відліків; $y_{\text{ф}}[k]$ — фактичне значення параметра; $y_{\text{цд}}[k]$ — результат цифрового двійника.

Мале значення похибки на одній вибірці ще не гарантує достовірності для всіх режимів.

Наприклад, цифровий двійник двигуна може добре працювати при навантаженні 30–70 % номінального, але не бути перевіреним біля теплової межі. У цьому випадку його не можна без додаткової валідації використовувати для прогнозування аварійного перевантаження.

Для цифрового двійника потрібно явно визначити:

- діапазон параметрів;
- перевірені режими;
- допустиму похибку;
- частоту оновлення;
- джерела даних;
- граничні умови застосування.

В огляді Sharma та співавторів одним із обмежень практичного впровадження цифрових двійників названо саме недостатню узгодженість визначень, архітектур та критеріїв оцінювання реальних реалізацій [3].

Тому правильне інженерне твердження звучить не «цифровий двійник точно відтворює об'єкт», а «цифровий двійник відтворює визначені властивості об'єкта з установленою точністю в заданому діапазоні режимів».

Висновки до лекції 10

Цифровий двійник є функціонально пов'язаним цифровим представленням фізичного виробничого об'єкта, яке використовує актуальні дані про його стан. Від звичайної цифрової моделі він відрізняється наявністю систематичного зв'язку з фізичним об'єктом та використанням цього зв'язку для моніторингу, прогнозування, аналізу або формування рішень.

Структура цифрового двійника може містити математичні, імітаційні, інформаційні та data-driven моделі. Їх склад визначається поставленою задачею, а не прагненням відтворити максимально можливу кількість властивостей об'єкта.

Синхронізація забезпечує відповідність цифрового стану фізичному. Розбіжність між ними може бути наслідком як зміни реального обладнання, так і похибки сенсорів або неадекватності самої моделі.

Цифровий двійник дозволяє оцінювати недоступні для прямого вимірювання параметри та прогнозувати подальшу поведінку виробничої системи. Зі збільшенням горизонту прогнозування невизначеність результату зазвичай зростає.

Віртуальне налагодження дозволяє перевірити значну частину програмної логіки до завершення фізичного монтажу обладнання. Його результат, однак, залежить від адекватності цифрової моделі і не скасовує фінальну перевірку на реальному об'єкті.

Достовірність цифрового двійника завжди обмежена конкретним діапазоном режимів і точністю вихідних даних. Тому цифровий двійник повинен мати не лише модель об'єкта, а й формально визначені межі її застосування.

Контрольні питання

1. Що розуміють під цифровою моделлю виробничого об'єкта?
2. Чим цифрова тінь відрізняється від цифрової моделі?
3. Яка ознака характеризує цифровий двійник?
4. Які компоненти можуть входити до структури цифрового двійника?
5. У чому полягає синхронізація фізичного та цифрового станів?
6. Про що може свідчити збільшення розбіжності між моделлю та об'єктом?
7. Які типи моделей можуть одночасно використовуватися у цифровому двійнику?
8. Для чого застосовуються гібридні фізичні та data-driven моделі?
9. Чому не всі параметри фізичного об'єкта вимірюються безпосередньо?
10. Як цифровий двійник може використовуватися для оцінювання стану?
11. Що розуміють під горизонтом прогнозування?
12. Чому зі збільшенням горизонту прогнозу зростає його невизначеність?
13. У чому полягає virtual commissioning?
14. Які алгоритми можна перевіряти методом віртуального налагодження?
15. Чому віртуальне налагодження не замінює повністю фізичних випробувань?
16. Які фактори обмежують достовірність цифрового двійника?
17. Чому область застосування цифрового двійника повинна бути формально визначена?

Список використаних джерел

1. Liu M., Fang S., Dong H., Xu C. Review of digital twin about concepts, technologies, and industrial applications. *Journal of Manufacturing Systems*. 2021. Vol. 58. P. 346–361. DOI: 10.1016/j.jmsy.2020.06.017.
2. Leng J., Wang D., Shen W., Li X., Liu Q., Chen X. Digital twins-based smart manufacturing system design in Industry 4.0: A review. *Journal of Manufacturing Systems*. 2021. Vol. 60. P. 119–137. DOI: 10.1016/j.jmsy.2021.05.011.
3. Sharma A., Kosasih E., Zhang J., Brintrup A., Calinescu A. Digital Twins: State of the art theory and practice, challenges, and open research questions. *Journal of Industrial Information Integration*. 2022. Vol. 30. Article 100383. DOI: 10.1016/j.jii.2022.100383.
4. ISO 23247-1:2021. *Automation systems and integration — Digital twin framework for manufacturing — Part 1: Overview and general principles*. Geneva : International Organization for Standardization, 2021.

5. Barbieri G., Bertuzzi A., Capriotti A. et al. A virtual commissioning based methodology to integrate digital twins into manufacturing systems. *Production Engineering*. 2021. Vol. 15. P. 397–412. DOI: 10.1007/s11740-021-01037-3.

6. Leng J., Zhou M., Xiao Y., Zhang H., Liu Q., Shen W., Su Q., Li L. Digital twins-based remote semi-physical commissioning of flow-type smart manufacturing systems. *Journal of Cleaner Production*. 2021. Vol. 306. Article 127278. DOI: 10.1016/j.jclepro.2021.127278.

7. He B., Bai K.-J. Digital twin-based sustainable intelligent manufacturing: a review. *Advances in Manufacturing*. 2021. Vol. 9. P. 1–21. DOI: 10.1007/s40436-020-00302-5.

8. Fu Y., Zhu G., Zhu M. et al. Digital Twin for Integration of Design-Manufacturing-Maintenance: An Overview. *Chinese Journal of Mechanical Engineering*. 2022. Vol. 35. Article 80. DOI: 10.1186/s10033-022-00760-x.

ЛЕКЦІЯ 11. ЗБИРАННЯ, ЗБЕРІГАННЯ ТА АНАЛІТИЧНЕ ОБРОБЛЕННЯ ВИРОБНИЧИХ ДАНИХ

Мета лекції

Мета лекції полягає у формуванні уявлення про життєвий цикл виробничих даних у кіберфізичних системах, способи їх збирання, синхронізації, зберігання та підготовки до аналітичного використання. Розглядаються потокові й історичні дані, часові ряди, промислові бази даних і historians, очищення та нормалізація даних, статистичні характеристики, кореляційний і регресійний аналіз, виявлення аномалій та застосування даних для оцінювання якості продукції і стану обладнання.

КФВС формує значні обсяги інформації з різних джерел: сенсорів, ПЛК, електроприводів, SCADA, MES, систем контролю якості та цифрових двійників. Саме накопичення цих даних не створює корисного знання. Для практичного застосування необхідно встановити їх фізичний зміст, часову відповідність, якість, структуру та зв'язок із конкретними технологічними режимами. Сучасні огляди промислової аналітики показують, що виробничі дані стають основою для моніторингу, прогнозування якості, технічного обслуговування та оптимізації лише після правильно організованого збирання і підготовки [1, 2].

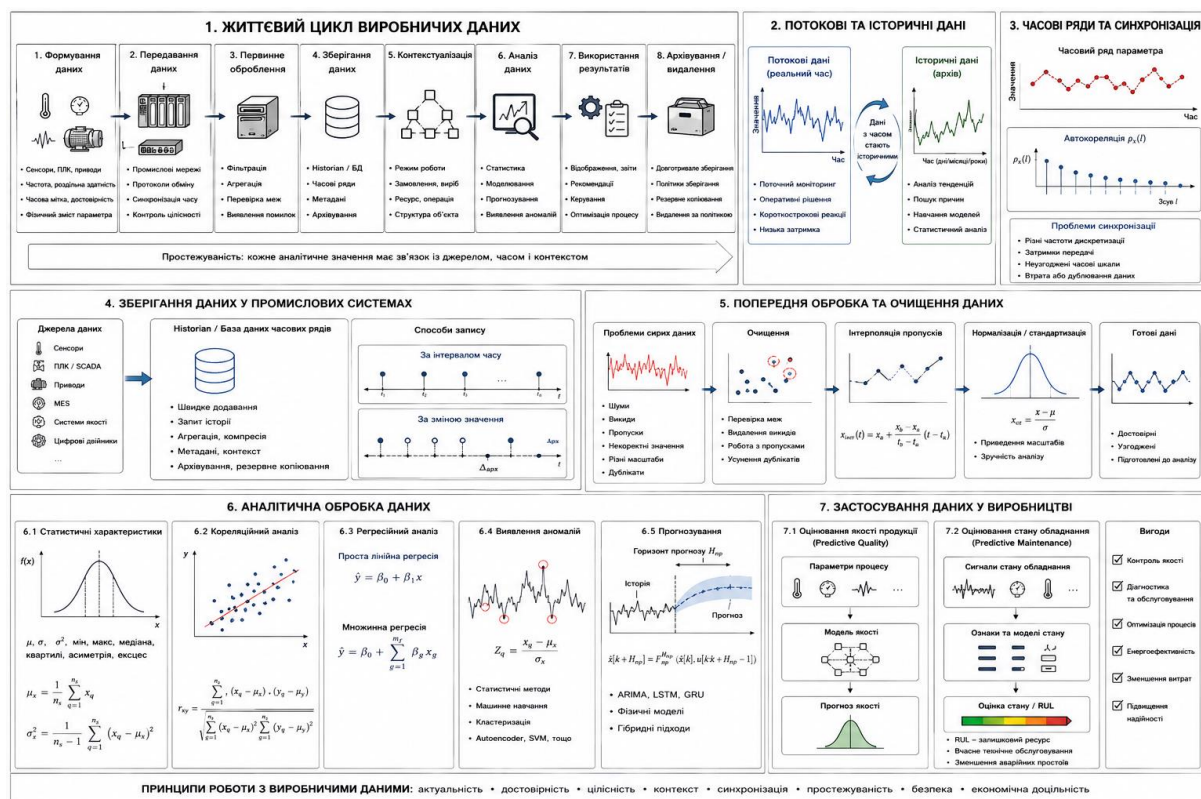


Рисунок 11.1 – Збирання, зберігання та аналітичне оброблення виробничих даних

11.1. Життєвий цикл виробничих даних

Виробничі дані проходять декілька послідовних стадій. У найзагальнішому вигляді життєвий цикл можна представити:

формування → передавання → первинне оброблення → зберігання → контекстуалізація → аналіз → використання результату → архівування або видалення.

Початковим джерелом можуть бути фізичний датчик, внутрішня змінна ПЛК, діагностичний параметр привода, результат контролю якості або запис MES.

На стадії формування даних визначаються:

- фізична величина;
- частота отримання;
- роздільна здатність;
- часова мітка;
- джерело;
- ознаки достовірності.

На наступних стадіях дані можуть перетворюватися. Наприклад, первинний вібраційний сигнал переходить у спектральні характеристики, а окремі технологічні сигнали — в узагальнений показник стану обладнання.

При цьому необхідно зберігати **простежуваність**. Аналітичний результат повинен мати зв'язок із даними, з яких він був отриманий, а ті — з конкретним фізичним об'єктом і моментом часу.

Повноту набору можна оцінити:

$$K_{\text{повн}} = \frac{N_{\text{кор}}}{N_{\text{оч}}},$$

де $N_{\text{кор}}$ — кількість коректно отриманих записів; $N_{\text{оч}}$ — очікувана кількість записів.

Значення, близьке до одиниці, означає, що більша частина очікуваних даних наявна. Проте високий коефіцієнт повноти не гарантує правильності самих значень.

У промислових системах проблема якості даних виникає вже на рівні їх формування. Неузгоджені часові шкали, різні частоти збирання та затримки різних пристроїв можуть створити неправильне взаємне розташування сигналів у часі [3].

11.2. Потоківі та історичні дані

За характером використання виробничі дані доцільно розділити на **потоківі та історичні**.

Потокові дані надходять у систему під час роботи виробничого процесу і повинні оброблятися без значного накопичення. Приклади:

- поточна температура;
- струм двигуна;
- швидкість конвеєра;
- стан клапана;
- повідомлення про аварію;
- поточний виробничий цикл.

Такі дані використовуються для оперативного моніторингу, виявлення відхилень і короткострокових рішень.

Історичні дані накопичуються протягом тривалого часу. Вони потрібні для:

- аналізу тенденцій;
- порівняння режимів;
- пошуку причин відмов;
- побудови статистичних моделей;
- прогнозування;
- навчання алгоритмів машинного навчання.

Між цими двома групами немає абсолютної межі. Потоківі значення після збереження стає історичним записом.

Для потоку даних можна визначити середню інтенсивність записів:

$$\lambda_d = \frac{N_{\text{зап}}}{T_{\text{спост}}},$$

де $N_{\text{зап}}$ — кількість записів за період спостереження; $T_{\text{спост}}$ — тривалість цього періоду.

При великій кількості сенсорів та високій частоті дискретизації обсяг історії швидко зростає. Тому необхідно вирішувати, які сигнали зберігати повністю, які агрегувати, а які записувати лише при зміні або виникненні події.

Аналітика великих виробничих масивів передбачає саме такий перехід від безперервного потоку сенсорних даних до структурованої інформації, що може використовуватися для прийняття рішень [1].

11.3. Часові ряди технологічних параметрів

Більша частина сенсорних даних КФВС являє собою **часові ряди** — послідовності вимірювань, впорядковані за часом.

Прикладами є:

- температура двигуна;
- віброшвидкість підшипника;
- тиск у трубопроводі;
- швидкість шпинделя;
- потужність установки;
- рівень у резервуарі.

Особливість часового ряду полягає у тому, що порядок відліків має фізичний зміст. Перестановка значень змінює інформацію про процес.

Для часових рядів характерна автокореляція — залежність поточного значення від попередніх.

Нормований коефіцієнт автокореляції для часового зсуву можна записати:

$$\rho_x(l) = \frac{\sum_{q=l+1}^{n_s} (x_q - \mu_x)(x_{q-l} - \mu_x)}{\sum_{q=1}^{n_s} (x_q - \mu_x)^2},$$

де l — кількість кроків часового зсуву; n_s — кількість відліків; μ_x — середнє значення ряду.

Автокореляція дозволяє виявляти інерційність, повторюваність і періодичні процеси.

Виробничі часові ряди часто є багатовимірними. Наприклад, стан електродвигуна одночасно описується струмом, швидкістю, температурою

та вібрацією. Взаємодія між такими рядами може містити більше інформації, ніж кожний сигнал окремо.

Огляд методів розпізнавання часових рядів у smart manufacturing показує їх застосування для класифікації режимів, контролю якості, діагностування та пошуку характерних шаблонів [4].

Окремою проблемою є синхронізація. Якщо струм і вібрація записані різними пристроями з неузгодженими годинниками, їх пряме порівняння може створити хибну причинну залежність [3].

11.4. Промислові бази даних та historians

Звичайна реляційна база даних добре підходить для замовлень, специфікацій, користувачів та інших структурованих записів. Для високочастотних технологічних параметрів використовуються спеціалізовані системи зберігання часових рядів, які в промисловій автоматизації часто називають **process historians** або просто historians.

Historian орієнтований на:

- запис значень із часовими мітками;
- швидке додавання нових відліків;
- отримання історії параметра за період;
- агрегацію;
- побудову трендів;
- архівування;
- зв'язок зі SCADA й аналітичними системами.

Не всі сигнали доцільно записувати при кожному циклі ПЛК. Якщо параметр змінюється повільно, застосовується запис за зміною або після певного інтервалу.

При записі за зміною нове значення зберігається, якщо:

$$|x[k] - x_{\text{арх}}| > \Delta_{\text{арх}},$$

де $x_{\text{арх}}$ — останнє збережене значення; $\Delta_{\text{арх}}$ — заданий поріг архівування.

Такий спосіб суттєво зменшує обсяг даних для повільних параметрів, але може бути непридатним, якщо для подальшого аналізу потрібна рівномірна часова сітка.

Для довготривалої аналітики необхідно зберігати не лише значення, але й контекст:

- режим роботи;
- номер виробу;
- виробниче замовлення;
- стан обладнання;

- версію рецепту;
- час калібрування сенсора.

Без цього через кілька місяців числовий ряд може втратити значну частину фізичного змісту. Дослідження промислової аналітики показують, що процесні дані стають корисними лише за поєднання з метаданими та знаннями про технологічний процес [5].

11.5. Попередня обробка й очищення даних

Первинні дані майже ніколи не є безпосередньо готовими до статистичного аналізу або навчання моделі. У них можуть міститися:

- пропуски;
- одиничні викиди;
- шум;
- дублювати;
- неправильні часові мітки;
- значення поза фізичними межами;
- різні масштаби змінних.

Сучасні огляди preprocessing часових рядів відносять до підготовки даних не тільки очищення, але й нормалізацію, зменшення розмірності, оброблення пропусків, компресію та інші операції [6].

Перший етап — **фізична перевірка**. Наприклад, від’ємна абсолютна температура або швидкість, що перевищує фізично можливу межу механізму, повинна бути позначена як некоректна.

Пропущені значення можуть:

- залишатися пропусками;
- видалятися;
- замінюватися попереднім значенням;
- інтерполюватися;
- оцінюватися за іншими параметрами.

Для лінійної інтерполяції між двома наявними точками:

$$x_{\text{інт}}(t) = x_a + \frac{x_b - x_a}{t_b - t_a}(t - t_a),$$

де $x_{\text{інт}}(t)$ — інтерпольоване значення; x_a , x_b — наявні значення на межах пропуску; t_a , t_b — відповідні моменти часу.

Такий спосіб допустимий для відносно плавного процесу й короткого пропуску. Для аварійно змінного сигналу інтерполяція може створити значення, яких фізично ніколи не існувало.

Для приведення параметрів різних масштабів часто застосовується стандартизація:

$$x_{\text{ст},q} = \frac{x_q - \mu_x}{\sigma_x},$$

де $x_{\text{ст},q}$ — стандартизоване значення; σ_x — стандартне відхилення.

Після такого перетворення змінні з різними фізичними масштабами можуть порівнюватися в одному статистичному алгоритмі.

Попередня обробка не повинна виконуватися механічно. Надмірна фільтрація може видалити саме ту короточасну зміну, яка є ознакою несправності.

11.6. Статистичні характеристики технологічних процесів

Першим рівнем аналітики є описова статистика. Вона дозволяє кількісно характеризувати типовий рівень параметра, його розкид і форму розподілу.

Середнє значення визначається:

$$\mu_x = \frac{1}{n_s} \sum_{q=1}^{n_s} x_q.$$

Дисперсія:

$$\sigma_x^2 = \frac{1}{n_s - 1} \sum_{q=1}^{n_s} (x_q - \mu_x)^2.$$

Крім середнього і дисперсії використовуються:

- медіана;
- мінімум і максимум;
- квантілі;
- коефіцієнт варіації;
- асиметрія;
- ексцес.

Для виробничого процесу важливо аналізувати не тільки значення статистик, а їх зміну в часі. Якщо середня температура залишається допустимою, але її дисперсія поступово зростає, це може свідчити про нестабільність режиму.

Варіабельність відносно середнього можна оцінити коефіцієнтом:

$$C_{\text{var}} = \frac{\sigma_x}{|\mu_x|},$$

де C_{var} — коефіцієнт варіації.

Його використання доцільне, якщо середнє не наближається до нуля.

Статистичні характеристики можуть розраховуватися не за всією історією, а у ковзному часовому вікні. Це дозволяє спостерігати повільну зміну властивостей обладнання.

Виробничі дані при цьому не завжди мають нормальний розподіл, тому застосування правила «три стандартні відхилення» без перевірки розподілу може бути некоректним.

11.7. Кореляційний та регресійний аналіз

Кореляційний аналіз використовується для оцінювання статистичного зв'язку між параметрами.

Коефіцієнт лінійної кореляції Пірсона можна записати:

$$r_{xy} = \frac{\sum_{q=1}^{n_s} (x_q - \mu_x)(y_q - \mu_y)}{\sqrt{\sum_{q=1}^{n_s} (x_q - \mu_x)^2 \sum_{q=1}^{n_s} (y_q - \mu_y)^2}},$$

де r_{xy} — коефіцієнт лінійної кореляції між параметрами; μ_y — середнє значення другого параметра.

r_{xy} лежить у межах від -1 до 1 . Значення, близьке за модулем до одиниці, характеризує сильний лінійний статистичний зв'язок.

Кореляція, однак, **не доводить причинність**. Два параметри можуть одночасно змінюватися через третю величину. Наприклад, температура двигуна та споживана потужність можуть корелювати через зміну механічного навантаження.

Для кількісного опису залежності застосовують регресію.

Найпростіша лінійна модель:

$$\hat{y}_q = \beta_0 + \beta_1 x_q,$$

де β_0 — вільний член регресії; β_1 — коефіцієнт нахилу регресійної залежності.

Для декількох факторів:

$$\hat{y}_q = \beta_0 + \sum_{g=1}^{m_f} \beta_g x_{g,q},$$

де m_f — кількість факторів; β_g — коефіцієнт відповідного фактора; $x_{g,q}$ — його значення.

Регресійна модель може використовуватися для оцінювання витрати енергії, якості поверхні, температури або іншого параметра за доступними технологічними змінними.

Для predictive quality саме зв'язок між процесними параметрами та вимірними характеристиками готового виробу формує основу data-driven прогнозування якості [7].

11.8. Виявлення аномальних станів

Аномалією називають спостереження або режим, що істотно відрізняється від нормальної поведінки системи.

Аномалія не обов'язково означає несправність. Причиною може бути:

- новий технологічний режим;
- зміна продукту;
- неправильне вимірювання;
- короткочасне збурення;
- деградація обладнання;
- реальна аварійна ситуація.

Найпростішим статистичним показником відхилення є стандартизована оцінка:

$$Z_q = \frac{x_q - \mu_x}{\sigma_x}$$

де Z_q — стандартизована міра відхилення від середнього.

Велике за модулем значення вказує на нетиповість точки відносно прийнятого статистичного розподілу. Проте такий критерій працює погано для багатовимірних та нелінійних процесів.

Для складніших задач застосовуються:

- багатовимірні статистичні методи;
- кластеризація;
- метод опорних векторів;
- дерева рішень;
- autoencoder;
- нейронні мережі;
- моделі часових рядів.

Огляд методів anomaly detection для smart manufacturing показує, що вибір алгоритму суттєво залежить від типу даних, наявності розмічених несправностей та структури нормальних режимів [8].

Практично складною є ситуація, коли аварійних прикладів мало. На добре обслуговуваному обладнанні реальні відмови рідкісні, тому класичне навчання за великою кількістю прикладів кожної несправності часто неможливе. Ця проблема також характерна для predictive maintenance [9].

Тому система виявлення аномалій повинна не просто видавати сигнал «аномалія», а зберігати контекст: режим обладнання, навантаження, продукт, час та інші супутні параметри.

11.9. Використання даних для оцінювання якості та стану обладнання

Підготовлені виробничі дані можуть використовуватися у двох близьких, але різних напрямках:

- оцінювання якості продукції;

- оцінювання технічного стану обладнання.

У predictive quality цільовою величиною є характеристика продукту: розмір, шорсткість, міцність, наявність дефекту або інший показник. Процесні параметри використовуються для прогнозування якості ще до завершення або безпосереднього вимірювання готового виробу [7].

Структурно:

технологічні параметри → модель → оцінка якості → рішення.

Для технічного стану схема інша:

вібрація, струм, температура, навантаження → ознаки → оцінка стану → прогноз → технічне обслуговування.

Predictive maintenance використовує сенсорні дані та аналітичні методи для визначення моменту, коли обслуговування доцільно виконати до виникнення функціональної відмови. Основними практичними проблемами є шум і помилки сенсорів, великі потоки даних та обмежена кількість прикладів реальних несправностей [9].

Точність бінарного алгоритму класифікації можна оцінити:

$$Acc = \frac{N_{\text{прав}}}{N_{\text{тест}}},$$

де A_{cc} — частка правильно класифікованих випадків; $N_{\text{прав}}$ — кількість правильних рішень; $N_{\text{тест}}$ — загальна кількість перевірочних прикладів.

Однак для рідкісних несправностей однієї точності недостатньо. Якщо 99 % даних відповідають справному стану, алгоритм, який завжди прогнозує «справно», матиме 99 % точності, але нульову діагностичну цінність для несправностей. Тому в таких задачах додатково аналізують повноту виявлення, precision, F1 та матрицю помилок.

Data-driven модель також повинна перевірятися в режимах, для яких вона призначена. Якщо модель навчалася лише при номінальному навантаженні, її використання при пуску або перевантаженні без додаткової перевірки може бути некоректним.

Отже, виробничі дані набувають цінності не через їх кількість, а через правильно сформований ланцюг:

достовірне вимірювання → синхронізація → контекст → очищення → аналіз → перевірена модель → інженерне рішення.

Висновки до лекції 11

Виробничі дані проходять повний життєвий цикл від формування у фізичному об'єкті до архівування та використання в аналітичних моделях.

На кожній стадії можуть виникати похибки, пропуски або втрата контексту, тому якість аналітики визначається не лише математичним методом.

Потокові дані використовуються для оперативної реакції, тоді як історичні масиви формують основу для аналізу тенденцій, прогнозування та навчання моделей. Значна частина промислових даних має структуру часових рядів, для яких принциповими є порядок відліків і синхронізація різних джерел.

Historian забезпечує спеціалізоване зберігання технологічної історії, але дані повинні супроводжуватися технологічним контекстом. Сам числовий ряд без інформації про режим роботи, виріб або стан обладнання має обмежену аналітичну цінність.

Попередня обробка включає перевірку фізичних меж, роботу з пропущеними значеннями, викидами, шумом та різними масштабами змінних. Спосіб очищення повинен враховувати фізику процесу, оскільки надмірна фільтрація може приховати реальну несправність.

Статистичний, кореляційний та регресійний аналіз дозволяють виявляти закономірності, однак статистичний зв'язок не є автоматичним доказом причинності.

Виявлення аномалій, predictive quality і predictive maintenance використовують спільну інформаційну основу, але розв'язують різні задачі. Для отримання достовірного результату модель повинна бути перевірена на незалежних даних та у тих режимах, для яких передбачається її застосування.

Контрольні питання

1. Які стадії утворюють життєвий цикл виробничих даних?
2. Що характеризує повнота набору даних?
3. Чим потокові дані відрізняються від історичних?
4. Що називають часовим рядом?
5. Який фізичний зміст має автокореляція?
6. Чому необхідно синхронізувати сигнали різних сенсорів?
7. Яке призначення process historian?
8. Чому не всі сигнали доцільно архівувати з максимальною частотою?
9. Які проблеми містять первинні виробничі дані?
10. Коли допустима інтерполяція пропущених значень?
11. Для чого застосовується стандартизація параметрів?
12. Які статистичні характеристики використовуються для аналізу технологічного процесу?
13. Що характеризує коефіцієнт кореляції?
14. Чому кореляція не доводить причинного зв'язку?
15. Для чого використовується регресійна модель?

16. Що розуміють під аномальним станом?
17. Чим аномалія відрізняється від підтвердженої несправності?
18. Чим predictive quality відрізняється від predictive maintenance?
19. Чому звичайної асигнації недостатньо для оцінювання алгоритму виявлення рідкісних несправностей?

Список використаних джерел

1. Wang J., Xu C., Zhang J., Zhong R. Big data analytics for intelligent manufacturing systems: A review. *Journal of Manufacturing Systems*. 2022. Vol. 62. P. 738–752. DOI: 10.1016/j.jmsy.2021.03.005.
2. Arruda H. M., Bavaresco R. S., Kunst R., Bugs E. F., Pesenti G. C., Barbosa J. L. V. Data Science Methods and Tools for Industry 4.0: A Systematic Literature Review and Taxonomy. *Sensors*. 2023. Vol. 23, No. 11. Article 5010. DOI: 10.3390/s23115010.
3. Schmetz A., Lee T. H., Zontar D., Brecher C. The Time Synchronization Problem in data-intensive Manufacturing. *Procedia CIRP*. 2022. Vol. 107. P. 827–832. DOI: 10.1016/j.procir.2022.05.070.
4. Farahani M. A., McCormick M. R., Gianinny R., Hudachek F., Harik R., Liu Z., Wuest T. Time-series pattern recognition in Smart Manufacturing Systems: A literature review and ontology. *Journal of Manufacturing Systems*. 2023. Vol. 69. P. 208–241. DOI: 10.1016/j.jmsy.2023.05.025.
5. Eger F., Reiff C., Brantl B., Colledani M., Verl A. Requirements towards optimizing analytics in industrial processes. *Procedia Computer Science*. 2021. Vol. 184. P. 597–605. DOI: 10.1016/j.procs.2021.03.074.
6. *Time-series data preprocessing: A survey and an empirical analysis*. *Journal of Engineering Research*. 2025. Vol. 13, No. 2. P. 674–711. DOI: 10.1016/j.jer.2024.02.018.
7. Tercan H., Meisen T. Machine learning and deep learning based predictive quality in manufacturing: a systematic review. *Journal of Intelligent Manufacturing*. 2022. Vol. 33. P. 1879–1905. DOI: 10.1007/s10845-022-01963-8.
8. Elía I., Pagola M. Anomaly detection in Smart-manufacturing era: A review. *Engineering Applications of Artificial Intelligence*. 2025. Vol. 139. Article 109578. DOI: 10.1016/j.engappai.2024.109578.
9. Santos J., Rocha E. M. Challenges in predictive maintenance – A review. *CIRP Journal of Manufacturing Science and Technology*. 2023. Vol. 40. P. 53–67. DOI: 10.1016/j.cirpj.2022.11.004.

ЛЕКЦІЯ 12. МАШИННЕ НАВЧАННЯ В КІБЕРФІЗИЧНИХ ВИРОБНИЧИХ СИСТЕМАХ

Мета лекції

Мета лекції полягає у формуванні уявлення про принципи застосування машинного навчання в кіберфізичних виробничих системах, підготовку навчальних даних, вибір ознак, побудову та перевірку моделей. Розглядаються навчання з учителем і без учителя, регресія, класифікація, кластеризація, нейронні мережі, виявлення аномалій, проблема перенавчання, метрики якості та інтерпретованість моделей.

Машинне навчання використовується у виробництві для прогнозування якості продукції, діагностування обладнання, оцінювання залишкового ресурсу, розпізнавання режимів, виявлення аномалій та прогнозування технологічних параметрів [1–4]. Воно не замінює фізичні моделі й автоматичне керування. Модель машинного навчання встановлює статистичні залежності на основі доступних даних, тому результат визначається структурою вибірки, режимами, представленими під час навчання, та стабільністю реального технологічного процесу

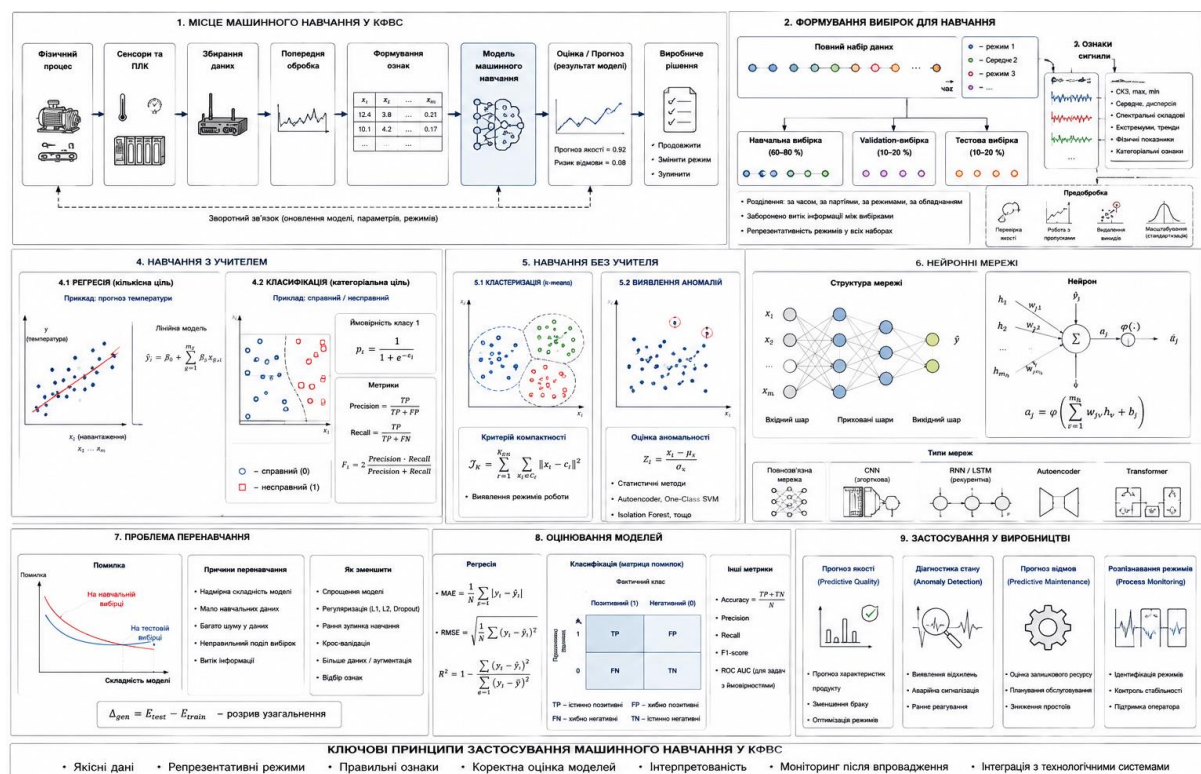


Рисунок 12.1 – Машинне навчання в кіберфізичних виробничих системах

12.1. Місце машинного навчання у структурі КФВС

Алгоритми машинного навчання працюють переважно на основі даних, сформованих сенсорними та інформаційними підсистемами КФВС. Типовий інформаційний ланцюг має вигляд:

фізичний процес → сенсори → збирання даних → попередня обробка → формування ознак → модель машинного навчання → оцінка або прогноз → виробниче рішення.

Модель може виконуватися на різних рівнях. Невеликий алгоритм класифікації може працювати на edge-пристрої. Аналіз значного історичного масиву або навчання глибокої нейронної мережі частіше виконується на серверному чи cloud-рівні.

У виробничій системі потрібно розрізняти **навчання моделі** та **використання навченої моделі**. Навчання може потребувати значних обчислювальних ресурсів і великого масиву історичних даних. Після навчання алгоритм нерідко виконує лише відносно компактну операцію прогнозування.

Набір даних для навчання можна представити:

$$\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N_{\text{ML}}},$$

де $\mathcal{D}$ — навчальний набір даних; N_{ML} — кількість об'єктів у вибірці; i — номер об'єкта вибірки.

Вектор $\mathbf{x}_i$ може містити температуру, тиск, струм, швидкість, вібраційні характеристики, тривалість операції або інші ознаки. Значення y_i задає цільову характеристику, якщо використовується навчання з учителем.

Наприклад, для predictive quality вхідними даними можуть бути параметри технологічного процесу, а цільовою змінною — фактична міцність або ознака придатності виробу. Саме поєднання процесних даних із результатами контролю якості створює основу для прогнозної моделі [2].

Машинне навчання не слід застосовувати лише тому, що підприємство накопичило великий масив даних. Спочатку потрібно сформулювати конкретну інженерну задачу: що необхідно прогнозувати, за який час до події, з якою допустимою похибкою і яке рішення буде прийнято за результатом прогнозу.

12.2. Формування навчальної та тестової вибірок

Для оцінювання моделі дані необхідно розділити щонайменше на навчальну та тестову частини.

$$\mathcal{D} = \mathcal{D}_{\text{train}} \cup \mathcal{D}_{\text{test}},$$

де $\mathcal{D}_{\text{train}}$ — дані для визначення параметрів моделі; $\mathcal{D}_{\text{test}}$ — незалежні дані для перевірки отриманої моделі.

Перетин цих вибірок не повинен містити ті самі об'єкти:

$$\mathcal{D}_{\text{train}} \cap \mathcal{D}_{\text{test}} = \emptyset.$$

Іноді додатково формується validation-вибірка, яка використовується для налаштування гіперпараметрів без звернення до фінального тестового набору.

Для виробничих часових рядів випадкове перемішування даних не завжди коректне. Якщо відліки одного технологічного циклу частково потрапили до навчальної, а частково до тестової вибірки, модель фактично перевіряється на інформації, дуже близькій до тієї, яку вона вже бачила.

Доцільніше розділяти дані:

- за виробничими партіями;
- за часовими періодами;
- за технологічними циклами;
- за конкретними виробами;
- іноді — за окремими машинами.

Для прогнозування майбутньої поведінки природним є часовий поділ: старіший період використовується для навчання, наступний — для перевірки.

Дані також повинні охоплювати реальний діапазон режимів. Якщо модель навчена лише при номінальному навантаженні обладнання, результат для пуску, перевантаження або іншого продукту не має автоматично вважатися достовірним.

Ця проблема особливо виражена для діагностування: експериментальні або модельні набори можуть містити добре представлені типові несправності, тоді як у реальному виробництві відмови рідкісні, режими змінюються, а дані мають шум [3].

12.3. Ознаки та підготовка даних для машинного навчання

Алгоритм працює не безпосередньо з фізичним процесом, а з набором числових ознак. Якість цих ознак часто впливає на результат сильніше, ніж вибір між двома близькими алгоритмами.

Ознаки можуть бути:

- безпосередніми вимірюваннями;
- статистичними характеристиками;
- спектральними параметрами;
- часовими характеристиками;
- розрахованими фізичними показниками;

- категоріальними ознаками технологічного режиму.

Наприклад, для вібраційної діагностики замість десятків тисяч відліків сигналу можна сформувати СКЗ, максимальне значення, коефіцієнт піковості та амплітуди окремих спектральних складових.

Для температурного процесу корисними можуть бути не тільки поточне значення, а й швидкість його зміни, середнє за певний інтервал та навантаження обладнання.

Перед навчанням виконують очищення, оброблення пропусків і масштабування, розглянуті в Лекції 11.

Надлишкова кількість ознак не завжди покращує модель. Частина параметрів може дублювати інформацію, бути випадковою або залежати від умов, які не повторюються під час експлуатації.

Нехай початково доступно m_0 параметрів, а після відбору залишається m_{sel} . Коефіцієнт скорочення простору ознак можна записати:

$$K_{\text{ознаки}} = \frac{m_{\text{sel}}}{m_0},$$

де $K_{\text{ознаки}}$ — частка залишених ознак; m_{sel} — кількість відібраних ознак; m_0 — їх початкова кількість.

Відбір ознак може виконуватися на основі фізичного аналізу, статистичних характеристик або спеціальних алгоритмів.

Для промислової задачі фізично обґрунтовані ознаки мають додаткову перевагу: інженеру простіше пояснити, чому модель змінила прогноз. Це особливо суттєво для відповідальних процесів, де проблема інтерпретованості моделей залишається одним із обмежень їх практичного використання [5, 6].

12.4. Навчання з учителем: регресія та класифікація

Навчання з учителем застосовується, коли для кожного навчального прикладу відоме правильне цільове значення.

Якщо ціль є числовою величиною, розв'язується задача регресії.

Приклади:

- прогноз температури;
- оцінювання енергоспоживання;
- прогноз шорсткості;
- визначення часу завершення операції;
- оцінювання залишкового ресурсу.

Загальну модель можна записати:

$$\hat{y}_i = f_{\theta}(\mathbf{x}_i),$$

де f_{θ} — модель машинного навчання з параметрами θ .

Одним із критеріїв навчання регресійної моделі є середньоквадратична функція втрат:

$$\mathcal{L}_{\text{MSE}} = \frac{1}{N_{\text{ML}}} \sum_{i=1}^{N_{\text{ML}}} (y_i - \hat{y}_i)^2,$$

Завдання навчання полягає в пошуку параметрів, за яких значення функції втрат є достатньо малим.

Якщо цільова змінна є категорією, розв'язується задача класифікації.

Наприклад:

- справний / несправний;
- придатний / брак;
- нормальний / аварійний режим;
- тип дефекту 1 / 2 / 3.

Класифікатор нерідко формує не лише номер класу, а оцінку ймовірності його належності.

Для двокласової задачі:

$$p_i = \frac{1}{1 + \exp(-s_i)},$$

де p_i — оцінена ймовірність належності до заданого класу; s_i — вихідний числовий показник класифікатора до перетворення у ймовірність.

Predictive quality використовує обидва типи задач. Модель може прогнозувати числове значення показника якості або безпосередньо визначати клас «придатно / непридатно» [2].

12.5. Навчання без учителя та кластеризація

У виробництві часто відсутня повна розмітка даних. Історія параметрів є, але оператор не вказував для кожного відліку точний стан машини.

У такому випадку застосовується навчання без учителя.

Алгоритм шукає структуру у самих даних:

- групи подібних режимів;
- нетипові об'єкти;
- приховані структури;
- взаємопов'язані ознаки.

Однією з основних задач є кластеризація.

Для методу k -means мінімізується критерій:

$$\mathcal{J}_K = \sum_{r=1}^{K_{\text{кл}}} \sum_{\mathbf{x}_i \in C_r} \|\mathbf{x}_i - \mathbf{c}_r\|^2,$$

де $\mathcal{J}_K$ — критерій компактності кластерів; $K_{\text{кл}}$ — кількість кластерів; C_r — множина об'єктів відповідного кластера; $\mathbf{c}_r$ — його центр.

У виробничій системі отримані кластери можуть відповідати різним технологічним режимам. Проте сам алгоритм не знає їх фізичного змісту. Після кластеризації інженер повинен встановити, чим реально відрізняються отримані групи.

Навчання без учителя корисне і для виявлення аномалій. Якщо більша частина історичних даних характеризує нормальний стан, алгоритм може сформувати область типової поведінки та позначати віддалені спостереження як нетипові.

Сучасні огляди anomaly detection у smart manufacturing показують одночасне застосування supervised, unsupervised і semi-supervised підходів; вибір залежить насамперед від доступності достовірних міток несправностей [7].

12.6. Нейронні мережі та глибоке навчання

Штучна нейронна мережа складається з взаємопов'язаних обчислювальних елементів — нейронів. Вона формує нелінійне відображення між входніми ознаками та результатом.

Вихід одного нейрона можна представити:

$$a_j = \varphi \left(\sum_{v=1}^{m_h} w_{jv} h_v + b_j \right),$$

де a_j — вихід нейрона; φ — функція активації; m_h — кількість його входів; w_{jv} — ваговий коефіцієнт; h_v — відповідний вхід нейрона; b_j — зміщення.

Об'єднання великої кількості таких елементів у декілька шарів формує глибоку нейронну мережу.

У виробничих задачах застосовуються:

- повнозв'язні мережі;
- convolutional neural networks;
- recurrent neural networks;
- LSTM;
- autoencoder;
- transformer-архітектури.

Convolutional networks природно використовувати для зображень і локальних закономірностей у сигналах. Рекурентні та LSTM-моделі застосовують до часових послідовностей. Autoencoder може навчатися відтворювати нормальні виробничі дані, після чого значна помилка реконструкції використовується як ознака аномалії.

Огляди predictive quality підтверджують активне використання deep learning у задачах прогнозування характеристик продукції та автоматичного контролю [2].

Глибока модель, однак, не завжди краща за простішу. Якщо даних мало, а залежність близька до лінійної, складна мережа може створити більше проблем із перенавчанням та інтерпретацією, ніж практичної користі.

Обчислювальну складність також потрібно враховувати при розміщенні моделі на edge-рівні. Алгоритм, який легко виконується на сервері, може не вкластися у допустимий час на промисловому контролері.

12.7. Навчання моделі та оптимізація її параметрів

Навчання полягає в зміні параметрів моделі так, щоб зменшити функцію втрат на навчальних даних.

Для багатьох моделей використовується градієнтний принцип:

$$\theta^{(r+1)} = \theta^{(r)} - \eta \nabla_{\theta} \mathcal{L},$$

де r — номер кроку оптимізації; η — швидкість навчання; ∇_{θ} — градієнт функції втрат за параметрами; $\mathcal{L}$ — обрана функція втрат.

Надто велике значення η може призводити до нестійкої зміни параметрів, а надто мале — до повільного навчання.

Крім параметрів, які визначаються під час навчання, існують гіперпараметри:

- структура моделі;
- кількість шарів;
- кількість нейронів;
- коефіцієнти регуляризації;
- розмір пакета даних;
- параметри конкретного алгоритму.

Їх підбирають за validation-даними або процедурами крос-валідації.

Промислова модель повинна оцінюватися не тільки за найкращою можливою точністю. Потрібно враховувати:

- стабільність результату;
- час виконання;
- потребу в пам'яті;
- простоту повторного навчання;
- можливість пояснення прогнозу;
- поведінку при відсутніх даних.

Огляд smart manufacturing показує, що AI/ML застосовується на різних стадіях виробництва, але інтеграція з реальними системами потребує

врахування не лише алгоритмічної точності, а й технічної інфраструктури [1].

12.8. Перенавчання та здатність моделі до узагальнення

Модель повинна не запам'ятати навчальні приклади, а правильно працювати на нових даних.

Перенавчання виникає, коли модель надто точно відтворює особливості навчальної вибірки, включаючи випадковий шум, але гірше працює на незалежних даних.

Різницю між помилками можна характеризувати:

$$\Delta_{\text{gen}} = E_{\text{test}} - E_{\text{train}},$$

де Δ_{gen} — розрив узагальнення; E_{test} — помилка на тестовій вибірці; E_{train} — помилка на навчальній вибірці.

Велике додатне значення може бути ознакою перенавчання.

Причинами є:

- надмірна складність моделі;
- мала кількість навчальних даних;
- велика кількість ознак;
- шум або помилки в розмітці;
- витік інформації між train і test;
- неадекватне представлення реальних режимів.

Для зменшення перенавчання застосовують:

- спрощення моделі;
- регуляризацію;
- ранню зупинку навчання;
- коректну крос-валідацію;
- збільшення репрезентативності вибірки;
- обґрунтований відбір ознак.

Для виробничих моделей існує ще одна проблема — зміна самого процесу. Обладнання зношується, змінюється матеріал, інструмент, режим, постачальник сировини. Модель може залишатися програмно незмінною, але статистичні властивості даних поступово відходять від навчальної вибірки.

Тому якість моделі повинна контролюватися і після введення в експлуатацію.

12.9. Метрики класифікації та проблема незбалансованих даних

Звичайна частка правильних рішень A_{cc} , введена в Лекції 11, не завжди достатня.

Для двокласової задачі використовують:

TP — правильно виявлені позитивні випадки;

TN — правильно визначені негативні;

FP — хибні позитивні;

FN — пропущені позитивні випадки.

Точність позитивних прогнозів:

$$Precision = \frac{TP}{TP + FP},$$

де TP — кількість істинно позитивних рішень; FP — кількість хибно позитивних рішень.

Повнота виявлення:

$$Recall = \frac{TP}{TP + FN},$$

де FN — кількість хибно негативних рішень.

Узгоджена оцінка precision та recall:

$$F_1 = 2 \frac{Precision \cdot Recall}{Precision + Recall},$$

де F_1 — гармонічна оцінка точності позитивного прогнозу та повноти виявлення.

Для діагностування ці показники мають зрозумілий фізичний зміст. Низький Recall означає, що модель пропускає значну частину несправностей. Низький Precision означає багато хибних тривог.

У промислових даних класи часто незбалансовані. Справний стан може становити 99 % історії, а відмови — лише 1 %. У такій ситуації висока загальна асигасу може маскувати нездатність алгоритму виявляти рідкісні стани.

Систематичний огляд реальних задач fault diagnosis підтверджує, що дефіцит даних про відмови та відмінність між лабораторними і реальними умовами є суттєвими проблемами практичного впровадження ML-діагностики [3].

12.10. Інтерпретованість і межі використання машинного навчання

Точний прогноз не завжди є достатнім для промислового рішення. Інженеру часто необхідно розуміти, чому модель вважає стан обладнання аварійним або чому прогнозує брак.

Це особливо потрібно, коли результат впливає на:

- зупинку обладнання;
- зміну технологічного режиму;

- відбракування продукції;
- планування ремонту;
- безпеку процесу.

Огляди interpretable machine learning у процесних галузях прямо вказують на проблему «чорної скриньки» як один із бар'єрів для використання складних моделей у відповідальних промислових задачах [5].

Для пояснення моделі можуть використовуватися:

- коефіцієнти регресії;
- важливість ознак;
- дерева рішень;
- локальні пояснення конкретного прогнозу;
- аналіз чутливості;
- візуалізація впливу параметрів.

Дослідження explainable AI для виробничих класифікаторів показують, що такі методи можуть допомагати оператору або інженеру співвіднести рішення моделі з реальною поведінкою процесу [6].

Проте пояснення алгоритму не доводить фізичної причинності. Якщо модель виявила сильний статистичний зв'язок між двома параметрами, це ще не означає, що один із них фізично викликає інший.

Для КФВС доцільно застосовувати машинне навчання там, де виконуються щонайменше такі умови:

- сформульована конкретна задача;
- наявні репрезентативні дані;
- визначена метрика якості;
- відокремлена незалежна перевірна вибірка;
- встановлена область допустимих режимів;
- передбачений контроль якості моделі після впровадження.

Виявлення аномалій, predictive quality і predictive maintenance є обґрунтованими напрямками застосування ML, але сучасні огляди показують значну різноманітність алгоритмів і відсутність одного методу, оптимального для всіх виробничих процесів [2, 7].

Висновки до лекції 12

Машинне навчання у КФВС використовується для встановлення закономірностей у виробничих даних та формування прогнозних або класифікаційних рішень. Його результат визначається не тільки вибором алгоритму, а якістю вимірювань, репрезентативністю вибірки та правильністю постановки задачі.

Навчання з учителем застосовується, якщо відомі цільові значення. Регресія використовується для числового прогнозу, класифікація — для

визначення стану або класу. Якщо достовірні мітки відсутні, застосовуються кластеризація, виявлення аномалій та інші методи навчання без учителя.

Нейронні мережі дозволяють моделювати складні нелінійні залежності, але потребують достатніх даних та обчислювальних ресурсів. Складніша модель не гарантує кращої роботи на реальному виробничому об'єкті.

Навчальна і тестова вибірки повинні бути незалежними. Для виробничих часових даних особливо небезпечним є витік інформації між вибірками через сусідні відліки одного технологічного циклу.

Перенавчання проявляється у високій якості на навчальних даних і погіршенні результату для нових режимів. Додатково під час експлуатації можуть змінюватися статистичні властивості самого технологічного процесу, тому модель потребує періодичного контролю.

Для рідкісних несправностей загальної асигнації недостатньо. Необхідно оцінювати precision, recall, F1 та аналізувати характер хибних рішень.

У відповідальних виробничих системах точність повинна доповнюватися інтерпретованістю, контролем області застосування та можливістю перевірити рішення за фізичними даними. Машинне навчання є частиною інженерної системи, а не автономною заміною знань про технологічний процес.

Контрольні питання

1. Які задачі машинного навчання розв'язуються у КФВС?
2. Чим навчання моделі відрізняється від використання навченої моделі?
3. Яке призначення навчальної вибірки?
4. Чому тестові дані повинні бути незалежними?
5. Чому випадкове перемішування часових виробничих даних не завжди коректне?
6. Що розуміють під ознакою машинного навчання?
7. У чому полягає задача регресії?
8. Для яких задач використовується класифікація?
9. Коли застосовується навчання без учителя?
10. У чому полягає принцип кластеризації?
11. Що являє собою штучний нейрон?
12. Для яких даних використовуються глибокі нейронні мережі?
13. Що називають функцією втрат?
14. Яке призначення швидкості навчання?
15. Що розуміють під перенавчанням?

16. Які причини можуть викликати перенавчання?
17. Що характеризують Precision та Recall?
18. Чому асигурація недостатньо для незбалансованих виробничих вибірок?
19. Для чого потрібна інтерпретованість моделі?
20. Чому статистичне пояснення прогнозу не є доказом фізичної причинності?

Список використаних джерел

1. Sahoo S., Lo C.-Y. Smart manufacturing powered by recent technological advancements: A review. *Journal of Manufacturing Systems*. 2022. Vol. 64. P. 236–250. DOI: 10.1016/j.jmsy.2022.06.008.
2. Tercan H., Meisen T. Machine learning and deep learning based predictive quality in manufacturing: a systematic review. *Journal of Intelligent Manufacturing*. 2022. Vol. 33. P. 1879–1905. DOI: 10.1007/s10845-022-01963-8.
3. Fernandes M., Corchado J. M., Marreiros G. Machine learning techniques applied to mechanical fault diagnosis and fault prognosis in the context of real industrial manufacturing use-cases: a systematic literature review. *Applied Intelligence*. 2022. Vol. 52. P. 14246–14280. DOI: 10.1007/s10489-022-03344-3.
4. Rajesh A. S. Smart Manufacturing through Machine Learning: A Review, Perspective, and Future Directions to the Machining Industry. *Journal of Engineering*. 2022. Vol. 2022. Article 9735862. DOI: 10.1155/2022/9735862.
5. Carter A., Imtiaz S., Naterer G. F. Review of interpretable machine learning for process industries. *Process Safety and Environmental Protection*. 2023. Vol. 170. P. 647–659. DOI: 10.1016/j.psep.2022.12.018.
6. Goldman C. V., Baltaxe M., Chakraborty D., Arinez J., Diaz C. E. Interpreting learning models in manufacturing processes: Towards explainable AI methods to improve trust in classifier predictions. *Journal of Industrial Information Integration*. 2023. Vol. 33. Article 100439. DOI: 10.1016/j.jii.2023.100439.
7. Elía I., Pagola M. Anomaly detection in Smart-manufacturing era: A review. *Engineering Applications of Artificial Intelligence*. 2025. Vol. 139. Article 109578. DOI: 10.1016/j.engappai.2024.109578.
8. Wang J., Xu C., Zhang J., Zhong R. Big data analytics for intelligent manufacturing systems: A review. *Journal of Manufacturing Systems*. 2022. Vol. 62. P. 738–752. DOI: 10.1016/j.jmsy.2021.03.005.

ЛЕКЦІЯ 13. КІБЕРБЕЗПЕКА КІБЕРФІЗИЧНИХ ВИРОБНИЧИХ СИСТЕМ

Мета лекції

Мета лекції полягає у формуванні системного уявлення про кібербезпеку кіберфізичних виробничих систем, характер загроз для операційних технологій та принципи побудови багаторівневого захисту. Розглядаються відмінності між безпекою IT- та OT-середовищ, моделі загроз, оцінювання кіберризиків, сегментація промислових мереж, концепція зон і каналів зв'язку ISA/IEC 62443, контроль доступу, захист віддаленого підключення, моніторинг мережевої та технологічної поведінки, резервне копіювання й реагування на інциденти.

Для КФВС кібербезпека має фізичний вимір. Компрометація інформаційного компонента може змінити команду ПЛК, уставку регулятора, стан виконавчого механізму або інформацію, за якою оператор приймає рішення. Через це захист виробничої системи не можна зводити до конфіденційності інформації. Необхідно одночасно враховувати безперервність технологічного процесу, цілісність керувальних дій, функціональну безпеку та можливість відновлення після інциденту [1–3]. NIST SP 800-82 Rev. 3 прямо розглядає захист OT з урахуванням специфічних вимог до продуктивності, надійності та safety.

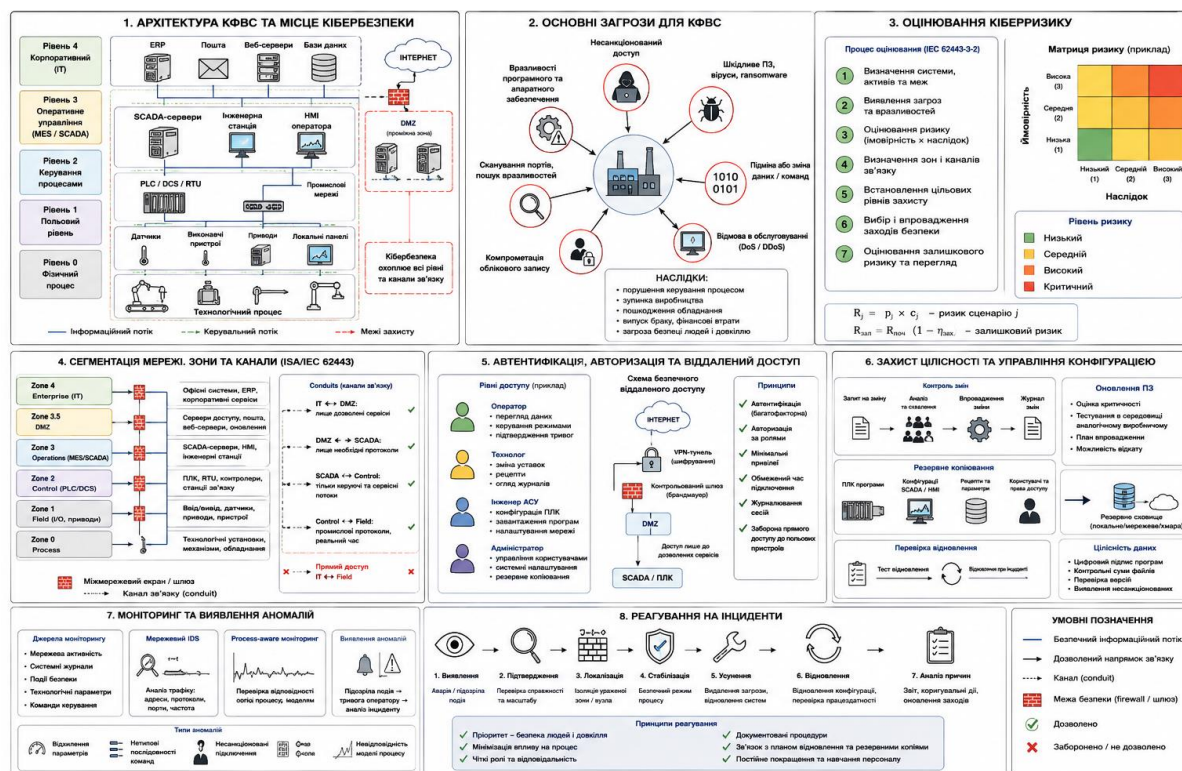


Рисунок 13.1 – Кібербезпека кіберфізичних виробничих систем

13.1. Особливості кібербезпеки операційних технологій

Інформаційна інфраструктура підприємства умовно поділяється на ІТ — Information Technology — та ОТ — Operational Technology.

ІТ-система переважно працює з інформаційними об'єктами: документами, обліковими даними, фінансовою інформацією, корпоративними застосунками та базами даних.

ОТ безпосередньо взаємодіє з фізичними процесами. До неї належать ПЛК, SCADA, DCS, приводи, промислові мережі, системи моніторингу та інші компоненти, які вимірюють або змінюють фізичний стан об'єкта. NIST використовує саме таке широке трактування ОТ і наголошує на її прямому зв'язку з фізичним середовищем [1].

Через це пріоритети захисту ІТ та ОТ частково відрізняються.

Для корпоративної ІТ-системи істотними є:

- конфіденційність інформації;
- цілісність даних;
- доступність сервісів.

Для виробничої ОТ-системи до них додаються:

- передбачуваність роботи;
- збереження керованості процесу;
- безпечний стан обладнання;
- мінімізація незапланованих зупинок;
- обмеження фізичних наслідків неправильної команди.

Наприклад, звичайне перезавантаження корпоративного комп'ютера для встановлення оновлення може бути допустимим. Перезавантаження ПЛК у момент виконання технологічної операції потребує аналізу наслідків.

Тому принцип «спочатку встановити оновлення, потім перевірити» для виробничої системи неприйнятний. Зміна програмного забезпечення повинна оцінюватися з точки зору сумісності, часових характеристик та впливу на фізичний процес.

Друга особливість — тривалий життєвий цикл. ІЕС 62443-2-1:2024 прямо враховує, що промислові системи можуть експлуатуватися понад двадцять років, а частина legacy-компонентів уже не підтримується виробником [2].

Отже, кібербезпека КФВС повинна працювати і в середовищі, де не кожен вразливість можна усунути заміною програмного забезпечення. У

таких випадках особливого значення набувають сегментація, обмеження доступу, моніторинг і компенсувальні заходи.

13.2. Основні властивості захищеної КФВС

Класична інформаційна безпека спирається на три базові властивості: конфіденційність, цілісність і доступність.

Конфіденційність означає, що інформація доступна лише авторизованим суб'єктам.

Для виробничих систем це можуть бути технологічні рецепти, конфігурація ПЛК, параметри обладнання, облікові записи або виробничі дані.

Цілісність означає, що інформація або команда не була несанкціоновано змінена.

У КФВС цілісність часто критичніша за конфіденційність. Якщо сторонній суб'єкт прочитає значення уставки температури, наслідки можуть бути обмеженими. Якщо ця уставка буде непомітно змінена, фізичний процес може перейти в недопустимий режим.

Доступність характеризує можливість використовувати необхідну функцію в потрібний момент.

Для ОТ це не лише доступність сервера. Потрібна працездатність повного ланцюга:

сенсор → мережа → контролер → виконавчий механізм.

До цих трьох властивостей у КФВС доцільно додати **автентичність** — можливість підтвердити, що команда або дані походять від очікуваного джерела.

Наприклад, пакет може бути формально правильним і своєчасно доставленим, але система повинна встановити, чи дійсно його сформував уповноважений контролер.

Кібербезпека не повинна руйнувати часові властивості системи. Якщо захисний механізм додає значну або нестабільну затримку у швидкому контурі керування, його реалізація потребує додаткового аналізу.

Тому захист КФВС є багатокритеріальним завданням:

security ↔ safety ↔ real-time ↔ availability.

Максимізація одного показника без урахування інших може погіршити систему в цілому.

13.3. Поверхня атаки та типові загрози

Поверхня атаки охоплює точки, через які може бути здійснений несанкціонований вплив на систему.

У сучасній КФВС до неї можуть належати:

- мережеві інтерфейси ПЛК;
- SCADA-сервери;
- робочі станції інженера;
- edge-шлюзи;
- IoT-пристрої;
- бездротові мережі;
- сервери віддаленого доступу;
- корпоративні інтеграційні інтерфейси;
- USB-носії;
- облікові записи користувачів;
- сторонні сервісні з'єднання.

Розвиток IoT та інтеграція IT і OT збільшують кількість таких точок. Огляди сучасної IoT-безпеки прямо пов'язують розширення функціональності й підключеності з появою нових загроз та вразливостей [4, 5].

Загрози можна класифікувати за характером впливу.

Несанкціоноване отримання інформації порушує конфіденційність.

Підміна або зміна даних впливає на їх цілісність.

Порушення доступності може призводити до втрати зв'язку або функцій керування.

Компрометація облікового запису дозволяє виконувати дії від імені легітимного користувача.

Несанкціонована зміна логіки керування небезпечна тим, що впливає безпосередньо на алгоритм фізичного процесу.

Для КФВС також характерні **process-aware** загрози. Вони враховують логіку технологічного процесу й можуть бути сформовані так, щоб мережеві повідомлення виглядали допустимими, але їх послідовність або значення створювали небажану фізичну поведінку. Систематичний огляд Rehman і Bahşi показує, що саме такі атаки стали окремим напрямом дослідження захисту ICS [7].

Це пояснює, чому звичайного мережевого моніторингу недостатньо. Необхідно також перевіряти фізичну логіку процесу.

13.4. Оцінювання кіберризiku

Не всі вразливості мають однакове практичне значення. Кіберризик визначається поєднанням можливості реалізації загрози та її наслідків.

Для набору сценаріїв спрощену очікувану оцінку ризику можна представити:

$$\mathcal{R}_{\text{кіб}} = \sum_{j=1}^{N_{\text{сц}}} p_j c_j,$$

де $\mathcal{R}_{\text{кіб}}$ — інтегральна оцінка кіберризiku; $N_{\text{сц}}$ — кількість розглянутих сценаріїв; p_j — оцінена ймовірність реалізації відповідного сценарію; c_j — оцінка його наслідків.

Це спрощена модель. На практиці значення p_j часто неможливо визначити як точну статистичну ймовірність, тому застосовують категоріальні або експертні шкали.

Для КФВС наслідок повинен оцінюватися не лише як вартість втрати інформації. Він може включати:

- зупинку виробництва;
- пошкодження обладнання;
- випуск некондиційної продукції;
- втрату простежуваності;
- порушення функцій безпеки;
- тривалий процес відновлення.

ІЕС 62443-3-2:2020 визначає формалізований процес оцінювання ризику для проєктування IACS. Він передбачає визначення System under Consideration, початкову оцінку ризику, поділ системи на zones and conduits, детальне оцінювання та встановлення цільових рівнів захисту для відповідних зон і каналів [3].

Після застосування захисних заходів залишається **залишковий ризик**.

Його умовно можна представити:

$$\mathcal{R}_{\text{зал}} = \mathcal{R}_{\text{поч}} (1 - \eta_{\text{зах}}),$$

де $\mathcal{R}_{\text{зал}}$ — залишковий ризик; $\mathcal{R}_{\text{поч}}$ — початкова оцінка ризику; $\eta_{\text{зах}}$ — умовна ефективність сукупності захисних заходів.

Це співвідношення є концептуальним, а не універсальним методом кількісного розрахунку ІЕС 62443. Воно показує, що захисні заходи не усувають ризик абсолютно.

Ризик потрібно повторно оцінювати після зміни архітектури, обладнання, способу віддаленого доступу або технологічного процесу. Огляд сучасних інструментів оцінювання ICS-ризиків також показує, що

кіберризик має розглядатися як змінна характеристика системи, а не одноразовий документ під час проєктування [8].

13.5. Сегментація мережі, зони та канали зв'язку

Одним із базових принципів промислової кібербезпеки є обмеження безпосередньої мережевої досяжності між компонентами.

Неправильна архітектура:

корпоративна мережа ↔ кожний ПЛК і привод.

У такій системі компрометація одного верхнього вузла може створити широкий шлях до технологічного обладнання.

Раціональніша структура використовує сегментацію:

IT → DMZ → OT-сервіси → керувальна мережа → польове обладнання.

ISA/IEC 62443 використовує поняття **zone** та **conduit**.

Зона об'єднує логічні або фізичні активи, для яких встановлюються спільні вимоги до безпеки.

Conduit визначає контрольований шлях комунікації між зонами.

IEC 62443-3-2:2020 вимагає поділяти System under Consideration на зони та канали, виконувати оцінювання ризику для них і визначати відповідні security requirements [3].

Приклад структури:

Enterprise zone → Industrial DMZ → SCADA/MES zone → Control zone → Safety/critical zone.

Межі зон можуть реалізовуватися міжмережевими екранами, промисловими шлюзами, контрольованими маршрутизаторами та іншими засобами.

При цьому VLAN сам по собі не потрібно автоматично вважати достатньою межею безпеки. Необхідно контролювати реальні дозволені потоки між сегментами.

Для кожного conduit доцільно визначити:

- які вузли можуть взаємодіяти;
- якими протоколами;
- у якому напрямку;
- які порти потрібні;
- чи дозволяється запис;
- чи потрібне журналювання.

Сегментація реалізує принцип мінімально необхідної зв'язаності: компоненти повинні мати доступ лише до тих ресурсів, які потрібні їх функції.

13.6. Автентифікація, авторизація та віддалений доступ

Автентифікація відповідає на питання:

хто намагається отримати доступ?

Авторизація визначає:

що цьому суб'єкту дозволено робити?

Ці два процеси необхідно розділяти. Користувач може бути правильно ідентифікований, але не мати права змінювати програму ПЛК.

У виробничій системі застосовується принцип **найменших привілеїв**. Користувач або сервіс отримує лише той обсяг доступу, який потрібний для виконання конкретної функції.

Наприклад:

- оператор — керування допустимими режимами;
- технолог — зміна частини уставок;
- інженер автоматизації — конфігурація ПЛК;
- адміністратор — системні функції.

Небажано використовувати один спільний обліковий запис для всього персоналу. У такому випадку неможливо встановити, хто виконав конкретну зміну.

Особливу проблему створює віддалений доступ. Він потрібний для сервісного обслуговування та діагностування, але одночасно формує зовнішній шлях до ОТ.

Рациональна схема передбачає:

- контрольовану точку входу;
- автентифікацію;
- обмеження прав;
- дозволений часовий інтервал;
- журналювання сесій;
- блокування прямого неконтрольованого доступу до польових пристроїв.

NIST SP 800-82 Rev. 3 рекомендує будувати захист ОТ відповідно до її специфічної архітектури й обмежувати мережеву взаємодію через контрольовані механізми [1].

Для legacy-пристроїв, які не підтримують сучасну автентифікацію, необхідно використовувати зовнішні компенсувальні заходи: сегментацію, контроль шлюзу та обмеження фізичного доступу.

13.7. Захист цілісності, оновлення, резервні копії та конфігурації

Стійка КФВС повинна мати можливість не тільки запобігати несанкціонованим змінам, а й виявляти їх і відновлювати попередню конфігурацію.

Критичними об'єктами є:

- програми ПЛК;
- параметри приводів;
- конфігурація SCADA;
- рецепти;
- мережеві налаштування;
- конфігурації edge-шлюзів;
- облікові записи.

Для них доцільно зберігати контрольовані версії.

Після модифікації повинно бути відомо:

хто → що → коли → навіщо змінив.

Така простежуваність суттєво полегшує відновлення після помилки або інциденту.

Оновлення програмного забезпечення також повинно бути керованим процесом. В ОТ не завжди можливо встановити виправлення відразу після публікації. Спочатку аналізуються:

- критичність вразливості;
- реальна доступність компонента для потенційного впливу;
- сумісність оновлення;
- необхідність зупинки;
- можливість відкату;
- компенсувальні заходи до моменту оновлення.

ІЕС 62443-2-1:2024 окремо враховує управління життєвим циклом та ситуації з legacy-системами, де встановлення актуальних патчів може бути технічно неможливим [2].

Резервне копіювання повинно охоплювати не лише інформаційну базу, а й конфігурацію засобів автоматизації.

Копія має практичну цінність тільки тоді, коли перевірена процедура її відновлення. Файл резервної програми ПЛК, який неможливо завантажити через відсутність потрібної версії програмного середовища, фактично не є повноцінним засобом відновлення.

13.8. Моніторинг і виявлення аномальної поведінки

Навіть добре спроектований превентивний захист не гарантує, що всі небажані події будуть заблоковані. Тому необхідний постійний моніторинг.

Можна виділити два рівні:

- **мережевий моніторинг;**
- **process-aware моніторинг.**

На мережевому рівні аналізуються:

- адреси;
- протоколи;
- частота з'єднань;
- нетипові маршрути;
- зміни конфігурації;
- поява нових вузлів.

Системи intrusion detection можуть застосовувати сигнатурні, поведінкові або ML-підходи. Сучасні огляди IIoT IDS відзначають активне застосування машинного та глибокого навчання, але також вказують на проблему обмежених ресурсів IIoT-вузлів, різноманіття протоколів та хибних спрацювань [6].

Для КФВС мережевого аналізу недостатньо.

Припустимо, контролер законно має право передавати клапану уставку. З точки зору мережі пакет може бути цілком правильним. Але якщо команда фізично суперечить поточному стану процесу, потрібна інша перевірка.

Process-aware моніторинг використовує модель технологічного процесу та перевіряє:

- фізичну допустимість значень;
- логіку послідовності команд;
- взаємозалежність сенсорів;
- відповідність команди реальному режиму.

Наприклад, при заявленому стані «двигун вимкнений» одночасне стійке зростання вимірної швидкості потребує перевірки незалежно від того, чи виглядає мережевий трафік нормальним.

Rehman і Bahşı показують, що process-aware security monitoring формує окремий напрямок захисту ICS саме через тісне поєднання кібернетичного і фізичного рівнів [7].

13.9. Реагування на кіберінциденти та відновлення

Захист повинен передбачати ситуацію, коли небажана подія вже відбулася.

Мета реагування полягає не лише у пошуку її джерела. Першочерговою є стабілізація виробничого процесу і недопущення розвитку фізичних наслідків.

Послідовність дій на концептуальному рівні:
**виявлення → підтвердження → локалізація → безпечний режим
→ усунення → відновлення → аналіз причин.**

Повний час до локалізації:

$$T_{\text{лок}} = T_{\text{дет}} + T_{\text{підтв}} + T_{\text{реак}},$$

де $T_{\text{лок}}$ — час від виникнення події до її локалізації; $T_{\text{дет}}$ — час виявлення; $T_{\text{підтв}}$ — час перевірки події; $T_{\text{реак}}$ — час виконання необхідної реакції.

Чим довше небажаний вплив залишається непоміченим, тим більша потенційна область його поширення.

Однак в ОТ автоматичне відключення мережевого вузла не завжди безпечне. Якщо він керує критичним технологічним механізмом, спочатку необхідно визначити допустиму фізичну реакцію.

Тому план реагування повинен містити не тільки ІТ-процедури, а й технологічні сценарії:

- продовження локальної автономної роботи;
- переведення на ручне керування;
- обмеження продуктивності;
- контрольоване завершення циклу;
- безпечне зупинення.

Після локалізації необхідно встановити довірену конфігурацію системи. Просте перезавантаження вузла недостатнє, якщо невідомо, чи не була змінена його програма.

Етап відновлення тому включає:

- перевірку конфігурації;
- відновлення з контрольованих копій;
- зміну компрометованих облікових даних;
- перевірку журналів;
- контроль фізичного стану об'єкта.

NIST SP 800-82 Rev. 3 розглядає заходи захисту ОТ в рамках повного управління ризиком, а ІЕС 62443-2-1:2024 визначає вимоги до програми кібербезпеки власника активів протягом експлуатації системи [1, 2].

Висновки до лекції 13

Кібербезпека КФВС відрізняється від захисту звичайних інформаційних систем тим, що результат кібернетичного впливу може проявлятися у фізичному технологічному процесі. Через це разом із конфіденційністю, цілісністю й доступністю необхідно враховувати керованість, часову передбачуваність і безпечну поведінку обладнання.

Підключення ПЛК, SCADA, PoT, edge- та корпоративних систем збільшує поверхню атаки. Захист повинен будуватися не шляхом ізоляції кожного окремого пристрою, а як властивість усієї архітектури.

Оцінювання кіберризиків дозволяє співвіднести загрозу з можливими фізичними та виробничими наслідками. IEC 62443-3-2:2020 задає ризик-орієнтований підхід до проєктування, включаючи визначення системи, зон і каналів та формування цільових вимог безпеки.

Мережева сегментація обмежує можливість неконтрольованого поширення впливу. Віддалений доступ повинен здійснюватися через контрольовані точки, із чіткою автентифікацією, авторизацією та журналюванням.

Захист системи включає контроль змін, управління оновленнями, резервне копіювання і перевірену процедуру відновлення. Для legacy-обладнання, де неможливо реалізувати сучасні механізми на самому пристрої, застосовуються компенсаторні заходи на рівні архітектури.

Моніторинг КФВС повинен аналізувати не тільки мережевий трафік, а й відповідність команд фізичній логіці процесу. Саме поєднання кібернетичних та технологічних ознак дозволяє виявляти небажані стани, які формально не порушують структуру мережевого протоколу.

Повністю безризикової КФВС не існує. Практична мета полягає у зменшенні ризику до прийняттого рівня та створенні системи, яка здатна своєчасно виявити небажаний вплив, локалізувати його і відновити керований виробничий процес.

Контрольні питання

1. Чим кібербезпека ОТ відрізняється від кібербезпеки корпоративної IT-системи?
2. Чому кіберінцидент у КФВС може мати фізичні наслідки?
3. Які властивості характеризують захищену виробничу систему?
4. Що розуміють під поверхнею атаки?
5. Які компоненти КФВС можуть створювати точки зовнішньої взаємодії?
6. Що таке process-aware загроза?
7. Від яких факторів залежить кіберризик?
8. Що розуміють під залишковим ризиком?
9. Яке призначення IEC 62443-3-2?
10. Що розуміють під зоною в ISA/IEC 62443?
11. Яку функцію виконує conduit?
12. Для чого застосовується сегментація промислової мережі?
13. Чим автентифікація відрізняється від авторизації?

14. У чому полягає принцип найменших привілеїв?
15. Чому віддалений доступ до ОТ потребує окремого контролю?
16. Які дані та конфігурації необхідно резервувати?
17. Чому встановлення оновлень в ОТ потребує попередньої перевірки?
18. Чим process-aware моніторинг відрізняється від звичайного мережевого IDS?
19. Чому автоматичне відключення скомпрометованого ОТ-вузла не завжди є безпечним рішенням?
20. Які етапи включає реагування на кіберінцидент?

Список використаних джерел

1. Stouffer K., Pease M., Tang C., Zimmerman T., Pillitteri V., Lightman S., Hahn A., Saravia S., Sherule A., Thompson M. *Guide to Operational Technology (OT) Security. NIST Special Publication 800-82 Revision 3*. Gaithersburg : National Institute of Standards and Technology, 2023. DOI: 10.6028/NIST.SP.800-82r3.
2. **IEC 62443-2-1:2024**. Security for industrial automation and control systems — Part 2-1: Security program requirements for IACS asset owners. Geneva : International Electrotechnical Commission, 2024.
3. **IEC 62443-3-2:2020**. Security for industrial automation and control systems — Part 3-2: Security risk assessment for system design. Geneva : International Electrotechnical Commission, 2020.
4. Alotaibi B. A Survey on Industrial Internet of Things Security: Requirements, Attacks, AI-Based Solutions, and Edge Computing Opportunities. *Sensors*. 2023. Vol. 23, No. 17. Article 7470. DOI: 10.3390/s23177470.
5. Mekala S. H., Baig Z., Anwar A., Zeadally S. Cybersecurity for Industrial IoT (IIoT): Threats, countermeasures, challenges and future directions. *Computer Communications*. 2023. Vol. 208. P. 294–320. DOI: 10.1016/j.comcom.2023.06.020.
6. Nuaimi M., Chaari Fourati L., Ben Hamed B. Intelligent approaches toward intrusion detection systems for Industrial Internet of Things: A systematic comprehensive review. *Journal of Network and Computer Applications*. 2023. Vol. 215. Article 103637. DOI: 10.1016/j.jnca.2023.103637.
7. Rehman M. ur, Bahşi H. Process-aware security monitoring in industrial control systems: A systematic review and future directions. *International Journal of Critical Infrastructure Protection*. 2024. Vol. 47. Article 100719. DOI: 10.1016/j.ijcip.2024.100719.
8. Rotibi A., Saxena N., Burnap P. Winning the battle with cyber risk identification tools in industrial control systems: A review. *IET Cyber-Physical Systems: Theory & Applications*. 2024. Vol. 9, No. 4. P. 350–365. DOI: 10.1049/cps2.12105.

ЛЕКЦІЯ 14. БАГАТОАГЕНТНІ СИСТЕМИ ТА ДЕЦЕНТРАЛІЗОВАНЕ КЕРУВАННЯ КІБЕРФІЗИЧНИМ ВИРОБНИЦТВОМ

Мета лекції

Мета лекції полягає у формуванні уявлення про принципи децентралізованого керування кіберфізичними виробничими системами та застосування багатоагентних систем для розподілу виробничих завдань, ресурсів і прийняття локальних рішень. Розглядаються властивості програмних агентів, структура багатоагентної КФВС, механізми взаємодії й координації, розподілене планування, переговорні та аукціонні процедури, самоорганізація, реконфігурація, багатоагентне навчання з підкріпленням та обмеження автономного керування.

Перехід від централізованої до розподіленої структури пов'язаний із розвитком КФВС, у яких окремі машини, роботи, транспортні модулі та інші виробничі ресурси вже мають власні засоби обчислення, зв'язку та отримання інформації про стан процесу. Це створює технічну можливість приймати частину рішень безпосередньо на рівні локального ресурсу. Багатоагентні системи є одним із підходів до реалізації такого принципу: фізичні та програмні компоненти представляються автономними агентами, які взаємодіють для досягнення локальних і загальних виробничих цілей [1–3]. Сучасний огляд агентного виробництва показує застосування цього підходу до планування, диспетчеризації, контролю, діагностування та реконфігурації CPPS, але водночас вказує на складність промислової валідації й передбачення поведінки розподілених систем.

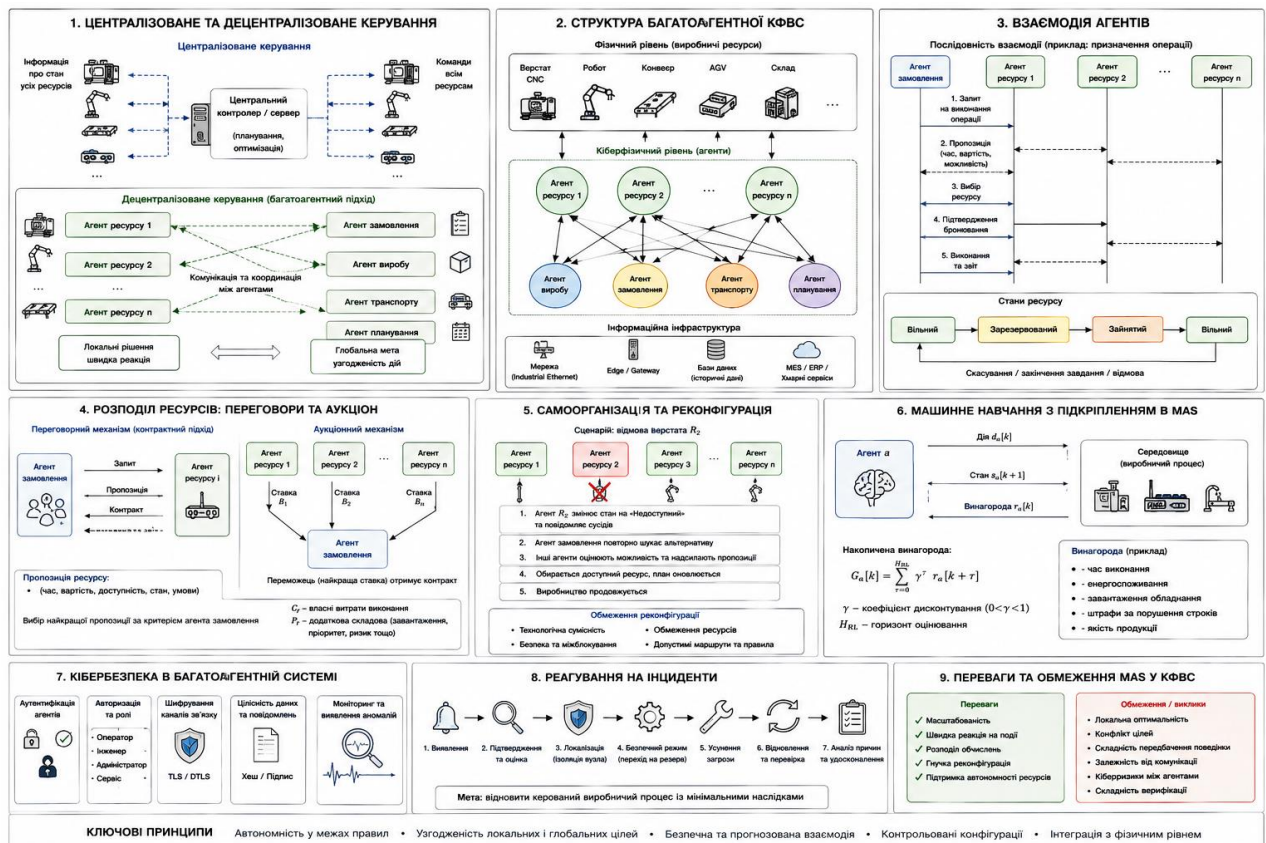


Рисунок 14.1 – Багатоагентні системи та децентралізоване керування кіберфізичним виробництвом

14.1. Централізоване та децентралізоване керування виробництвом

У централізованій системі основна інформація про стан виробництва передається до одного керувального вузла. Він формує план, розподіляє ресурси та передає команди локальним компонентам.

Така структура зручна, якщо:

- склад обладнання змінюється рідко;
- технологічні маршрути відомі заздалегідь;
- центральний вузол має достатню інформацію;
- час розрахунку не обмежує виробничу реакцію.

Перевагою централізованого підходу є можливість оптимізувати систему за глобальним критерієм. Центральний алгоритм потенційно бачить усі ресурси та всі замовлення.

Недолік проявляється при збільшенні масштабу й динаміки системи. Кожне локальне відхилення — відмова верстата, запізнення матеріалу, зміна пріоритету замовлення — повинно бути передане центру, після чого необхідно повторно сформулювати рішення.

У децентралізованій системі частина повноважень переноситься до локальних ресурсів. Машина, транспортний модуль або виробничий

осередок можуть самостійно оцінювати свій стан, приймати допустиме рішення та взаємодіяти з іншими компонентами.

Розподіл великої задачі на локальні можна представити:

$$\mathcal{P} = \bigcup_{a=1}^{N_{\text{ar}}} \mathcal{P}_a,$$

де $\mathcal{P}$ — загальна задача виробничого керування; N_{ar} — кількість агентів; $\mathcal{P}_a$ — локальна задача відповідного агента.

Такий поділ зменшує обсяг обчислень в одному центральному вузлі та дозволяє швидше реагувати на локальну подію. Водночас окремий агент бачить лише частину системи й може прийняти рішення, яке є добрим локально, але не оптимальним для виробництва в цілому. Саме цей компроміс між локальною реактивністю та глобальною якістю рішення є центральною проблемою децентралізації [2, 4]. Дослідження автономного виробничого керування вказують, що розподіл задачі прискорює локальне прийняття рішень, але обмежений інформаційний горизонт може приводити до локальних оптимумів; тому практичні архітектури часто залишають певний рівень централізованої координації.

14.2. Поняття виробничого агента

У контексті КФВС **агент** доцільно визначати як програмний або кіберфізичний компонент, який має власний стан, отримує інформацію про середовище, самостійно формує допустимі рішення та взаємодіє з іншими агентами для досягнення визначених цілей.

Для виробничого агента характерні такі властивості:

автономність — можливість приймати частину рішень без безпосередньої команди центрального контролера;

реактивність — здатність змінювати поведінку при зміні середовища;

проактивність — здатність не лише реагувати на подію, а й самостійно ініціювати дію для досягнення мети;

комунікаційність — можливість обмінюватися повідомленнями з іншими компонентами;

кооперативність — можливість узгоджувати локальні дії з іншими агентами.

Ці властивості систематично використовуються в сучасних роботах з agent-based manufacturing [1].

Виробничий агент не обов'язково відповідає одному фізичному пристрою. Можливі різні варіанти.

Resource Agent представляє виробничий ресурс: верстат, робот, транспортний модуль або ділянку.

Product Agent представляє виріб і може містити інформацію про потрібний технологічний маршрут.

Order Agent представляє виробниче замовлення та його пріоритет.

Transport Agent координує переміщення матеріалу.

Scheduler Agent виконує локальне або глобальне планування.

Наприклад, ресурсний агент верстата може знати його поточний стан, доступні операції, чергу замовлень, прогнозований час завершення роботи та діагностичні обмеження.

Важливо відрізнити **автономність агента** від повної незалежності. Виробничий агент працює в межах правил, заданих архітектурою. Він не повинен самостійно скасовувати вимоги безпеки, перевищувати технологічні обмеження або змінювати правила доступу до обладнання.

14.3. Структура багатоагентної виробничої системи

Багатоагентна система — Multi-Agent System, MAS — складається з декількох агентів, які спільно функціонують у певному середовищі та можуть мати як спільні, так і локальні цілі.

У КФВС фізичне обладнання і програмна агентна структура можуть бути пов'язані таким чином:

фізичний ресурс ↔ локальний контролер ↔ агент ресурсу ↔ мережа агентів.

ПЛК при цьому не обов'язково виконує саму агентну логіку. Він може забезпечувати детерміноване локальне керування, а агент — вирішувати виробничу задачу вищого рівня: приймати замовлення, оцінювати доступність або домовлятися про ресурс.

Такий розподіл має принципове значення. Швидкий контур електропривода не потрібно перетворювати на переговорний процес між агентами. Агент задає допустиму виробничу дію, а її фізична реалізація залишається за локальним керуванням.

Стан агента:

$$\mathbf{s}_a[k] = [\mathbf{s}_a^{\text{лок}}[k], \mathbf{s}_a^{\text{інф}}[k]],$$

де $\mathbf{s}_a[k]$ — інформаційний стан агента; $\mathbf{s}_a^{\text{лок}}[k]$ — відомості про власний ресурс; $\mathbf{s}_a^{\text{інф}}[k]$ — доступні йому відомості про інші компоненти системи.

На основі цього стану агент визначає дію:

$$d_a[k] = \pi_a(\mathbf{s}_a[k]),$$

де $d_a[k]$ — рішення агента; π_a — його політика прийняття рішень.

Локальна інформація не завжди містить повний стан виробництва. Саме тому результат залежить від того, які дані агенти передають один одному.

Чим більше інформації обмінюється між агентами, тим ближче локальні рішення можуть бути до глобально обґрунтованих. Але одночасно зростає мережеве навантаження та складність координації.

14.4. Комунікація та координація агентів

Багатоагентна система не може функціонувати як набір повністю незалежних програм. Агенти повинні обмінюватися інформацією про запити, пропозиції, стани й результати.

Типовий цикл взаємодії може мати вигляд:

запит → оцінювання → пропозиція → вибір → підтвердження → виконання → повідомлення про результат.

Наприклад, агент виробу повинен вибрати верстат для наступної операції. Він надсилає запит декільком ресурсним агентам. Кожен оцінює власну доступність і повертає пропозицію.

Пропозиція ресурсу може містити:

- можливість виконання операції;
- доступний час;
- прогнозовану тривалість;
- виробничу вартість;
- завантаження;
- технічний стан.

Після отримання відповідей агент замовлення формує рішення.

Для комунікації важливим є не лише транспорт повідомлення, а і його семантика. Повідомлення «resource available» повинно однаково трактуватися всіма учасниками.

Тому агентні архітектури потребують формалізованих моделей даних, ідентифікації активів і однозначного опису виробничих можливостей. У цьому аспекті MAS безпосередньо пов'язані з OPC UA, інформаційними моделями та семантичною інтероперабельністю, розглянутими в попередніх лекціях.

Координація особливо потрібна при конкуренції за один ресурс. Якщо два незалежні агенти одночасно вважають один верстат доступним, без процедури резервування виникне конфлікт.

Тому стан ресурсу може включати проміжний статус:

вільний → зарезервований → зайнятий → вільний.

Резервування повинно мати обмежений час. Інакше невиконана або втрачена транзакція може необґрунтовано заблокувати ресурс.

14.5. Розподілене призначення виробничих завдань

Однією з типових задач MAS є розподіл замовлень між доступними виробничими ресурсами.

Нехай агент має декілька можливих ресурсів і повинен обрати один із них. Для кожної альтернативи можна сформулювати локальну функцію вартості:

$$C_{a,r} = w_{\text{ч}} T_{a,r} + w_{\text{в}} V_{a,r} + w_{\text{з}} Z_r,$$

де $C_{a,r}$ — локальна оцінка використання ресурсу r агентом a ; $w_{\text{ч}}$, $w_{\text{в}}$, $w_{\text{з}}$ — вагові коефіцієнти часу, виробничих витрат і завантаження; $T_{a,r}$ — очікуваний час виконання; $V_{a,r}$ — оцінені витрати; Z_r — поточний рівень завантаження ресурсу.

Вибирається ресурс:

$$r^* = \arg \min_{r \in \mathcal{R}_a} C_{a,r},$$

де r^* — вибраний ресурс; $\mathcal{R}_a$ — множина ресурсів, здатних виконати потрібну операцію.

Це локальне рішення.

Якщо декілька агентів одночасно використовують однаковий критерій, вони можуть вибрати один і той самий ресурс. Після цього необхідний додатковий механізм координації.

У реактивній децентралізованій системі план може перебудовуватися при події:

- відмові машини;
- появи термінового замовлення;
- зміні часу оброблення;
- недоступності матеріалу.

Дослідження Okroti та Jeong демонструє саме подієво-керований децентралізований підхід, де виробничий план повторно оптимізується після динамічних подій; працездатність підходу перевірялася на прототипі smart factory [4].

14.6. Переговорні та аукціонні механізми

Для розподілу ресурсів між автономними агентами можуть використовуватися переговорні механізми.

Один із зрозумілих варіантів — контрактний принцип:

- агент замовлення оголошує задачу;

- доступні ресурси формують пропозиції;
- замовник порівнює пропозиції;
- найкращий ресурс отримує контракт;
- інші агенти звільняються від цього запиту.

Аукціонний підхід дозволяє розподіляти ресурси без одного централізованого плану, але правила аукціону фактично стають частиною алгоритму виробничого керування.

Ціну пропозиції можна сформулювати:

$$B_r = C_r + P_r,$$

де B_r — ставка агента ресурсу; C_r — власна оцінка витрат виконання; P_r — додаткова складова, яка враховує завантаження, пріоритет або інші виробничі умови.

Мінімальна ставка не завжди повинна перемагати. Наприклад, для термінового замовлення більш істотним може бути час, а для звичайного — вартість.

Тому правила переговорів повинні узгоджуватися із загальною виробничою метою.

Якщо локальні агенти оптимізують тільки власні витрати, система може отримати небажану поведінку. Ресурс може відмовлятися від короткостроково невигідної операції, хоча її виконання критичне для загального виробничого плану.

Звідси випливає одна з основних вимог до MAS: локальна функція винагороди або вартості повинна бути пов'язана з глобальною метою системи.

14.7. Самоорганізація та реконфігурація КФВС

Децентралізоване керування особливо корисне для реконфігурованого виробництва.

Припустимо, один верстат виходить із ладу. У жорстко централізованій структурі необхідно перебудувати план для значної частини системи.

У багатоагентній архітектурі можливий інший алгоритм:

- агент несправного ресурсу змінює стан на «недоступний»;
- повідомляє агентів, чиї операції були заплановані;
- відповідні агенти повторно шукають сумісний ресурс;
- формується новий локальний план;
- виробництво продовжується з доступними ресурсами.

Таку властивість називають самоорганізацією або локальною реконфігурацією.

Системи CPPS розглядаються як технологічна основа для децентралізованого виробництва саме завдяки локальним обчисленням, комунікації та можливості автономної реакції [3].

Проте самоорганізація не означає хаотичну зміну структури. Повинні залишатися незмінними:

- правила технологічної сумісності;
- вимоги безпеки;
- допустимі маршрути;
- права доступу;
- обмеження ресурсів.

Реконфігурація повинна виконуватися тільки в межах допустимого простору конфігурацій.

Його можна представити:

$$C_{\text{доп}} = \{c \in C : g_v(c) \leq 0, v = 1, \dots, N_g\},$$

де $C_{\text{доп}}$ — множина допустимих конфігурацій; C — множина всіх можливих конфігурацій; g_v — відповідне обмеження; N_g — кількість обмежень.

Цей принцип важливий при проектуванні автономної КФВС: система може самостійно вибирати рішення, але не повинна виходити за межі перевірених і безпечних конфігурацій.

14.8. Багатоагентне навчання з підкріпленням

У попередній лекції розглядалося машинне навчання за наявними виробничими даними. Інший підхід полягає в тому, що агент навчається через взаємодію із середовищем.

У **reinforcement learning** агент отримує стан, виконує дію та отримує винагороду.

Цикл має вигляд:

стан → дія → новий стан → винагорода → корекція політики.

У багатоагентній системі одночасно навчаються декілька агентів, тому середовище для кожного з них змінюється ще й через рішення інших учасників.

Сумарну дисконтовану винагороду можна записати:

$$G_a[k] = \sum_{\tau=0}^{H_{\text{RL}}} \gamma^{\tau} r_a[k + \tau],$$

де $G_a[k]$ — накопичена винагорода агента; H_{RL} — горизонт її оцінювання; γ — коефіцієнт дисконтування; r_a — поточна винагорода.

Значення γ , близьке до нуля, робить агента більш орієнтованим на короткостроковий результат. Більше значення збільшує вплив майбутніх наслідків.

У виробництві винагорода може враховувати:

- час виконання;
- простої;
- витрати енергії;
- порушення термінів;
- завантаження обладнання;
- якість.

Multi-agent reinforcement learning досліджується для розподілу виробничих задач та самооптимізації. Наприклад, Zheng та співавтори поєднали промисловий knowledge graph і multi-agent reinforcement learning для самооптимізованого виконання виробничих завдань [5].

Проте пряме навчання методом проб і помилок на реальному промисловому обладнанні може бути неприпустимим. Агент під час навчання неминуче робить невдалі дії.

Тому використовуються:

- імітаційні середовища;
- цифрові двійники;
- історичні дані;
- безпечні обмеження політики;
- попереднє офлайн-навчання.

Це безпосередньо пов'язує MAS із цифровими двійниками, розглянутими в Лекції 10.

14.9. Обмеження автономного та багатоагентного керування

Основна перевага MAS — розподіл інтелекту між компонентами — одночасно створює її основну проблему: загальна поведінка системи формується взаємодією багатьох локальних рішень.

Через це виникають такі обмеження.

Локальна оптимальність

Агент бачить обмежену частину виробництва. Оптимальне локальне рішення може погіршувати глобальний результат.

Конфлікт цілей

Один агент намагається мінімізувати час виконання замовлення, інший — енерговитрати, третій — завантаження ресурсу. Без загальної координації ці цілі можуть суперечити одна одній.

Складність передбачення поведінки

У централізованій системі порядок виконання здебільшого визначається єдиним алгоритмом. У MAS результат може виникати як **емергентна поведінка** внаслідок локальних взаємодій.

Огляд інструментів моделювання CPPS із розподіленим інтелектом прямо називає труднощі прогнозування продуктивності одним із бар'єрів широкого промислового впровадження таких структур [6].

Комунікаційна залежність

Автономність окремих агентів не усуває потреби у зв'язку. Втрата повідомлення може призвести до того, що два компоненти працюватимуть із різними уявленнями про стан ресурсу.

Кібербезпека

Компрометований агент може передавати формально правильні, але хибні пропозиції іншим учасникам. Тому міжагентна комунікація потребує автентифікації та контролю доступу.

Верифікація

Необхідно довести, що локальні правила не можуть утворити небезпечну або блокувальну конфігурацію.

З цієї причини для реального виробництва доцільна не абсолютна децентралізація, а **гібридна архітектура**.

Її можна представити так:

локальний детермінований контроль → автономні агенти → координувальний рівень → MES/виробничі цілі.

ПЛК забезпечують фізичну безпеку та швидке керування. Агенти приймають локальні виробничі рішення. Координувальний рівень контролює глобальні обмеження, а MES задає загальне виробниче завдання.

Такий підхід поєднує реактивність децентралізації з можливістю контролю глобальної поведінки. Огляди автономного production planning також показують, що питання полягає не у виборі лише централізованого або лише децентралізованого керування, а у визначенні доцільного ступеня розподілу повноважень [2].

Висновки до лекції 14

Децентралізоване керування переносить частину функцій прийняття рішень від центрального вузла до локальних кіберфізичних ресурсів. Технічною основою такого підходу є наявність у сучасних машин, роботів і контролерів власних обчислювальних, сенсорних і комунікаційних можливостей.

Багатоагентна система представляє виробничі компоненти як автономні програмні сутності. Агенти можуть представляти ресурси, вироби, замовлення, транспортні модулі та координаційні функції. Їх автономність обмежується технологічними, часовими та безпековими вимогами.

Розподілене призначення завдань дозволяє локально реагувати на відмови обладнання, зміни замовлень і доступності ресурсів. Для вирішення конфліктів використовуються переговорні, контрактні та аукціонні механізми.

Самоорганізація дозволяє реконфігурувати систему без повного централізованого перепланування, але можливі зміни повинні залишатися в межах формально визначеного простору допустимих конфігурацій.

Багатоагентне навчання з підкріпленням дозволяє агентам адаптувати політики на основі результатів взаємодії. Для промислового використання навчання доцільно виконувати переважно в імітаційному середовищі або цифровому двійнику, оскільки неконтрольований метод проб і помилок на реальному обладнанні неприйнятний.

Основною проблемою MAS є поява глобальної поведінки з великої кількості локальних рішень. Через це складніше прогнозувати продуктивність, доводити відсутність конфліктів і гарантувати глобальну оптимальність.

Для реальної КФВС найбільш обґрунтованою є гібридна структура: фізично критичні функції виконуються детермінованими локальними контролерами, агентний рівень забезпечує автономність і реконфігурацію, а верхній координувальний рівень контролює глобальні виробничі цілі.

Контрольні питання

1. Чим централізоване керування відрізняється від децентралізованого?
2. Які переваги має локальне прийняття рішень?
3. Чому локально оптимальне рішення не завжди є оптимальним для всієї системи?
4. Що розуміють під виробничим агентом?

5. Які властивості характерні для агента?
6. Які типи агентів можуть використовуватися у виробничій системі?
7. Чому автономність агента не означає його повної незалежності?
8. Як пов'язуються фізичний ресурс, ПЛК і програмний агент?
9. Які дані можуть передаватися між агентами?
10. Для чого потрібне резервування виробничого ресурсу?
11. Як агент може вибирати між декількома доступними ресурсами?
12. У чому полягає переговорний механізм розподілу завдань?
13. Які переваги аукціонного розподілу ресурсів?
14. Що розуміють під самоорганізацією КФВС?
15. Чому реконфігурація повинна мати формально визначені обмеження?
16. У чому полягає принцип reinforcement learning?
17. Чому навчання автономного агента на реальному обладнанні має суттєві обмеження?
18. Що розуміють під емергентною поведінкою багатоагентної системи?
19. Які фактори ускладнюють промислове впровадження MAS?
20. Чому гібридна архітектура часто доцільніша за повністю децентралізовану?

Список використаних джерел

1. Pulikottil T., Estrada-Jimenez L. A., Ur Rehman H., Mo F., Nikghadam-Hojjati S., Barata J. Agent-based manufacturing — review and expert evaluation. *The International Journal of Advanced Manufacturing Technology*. 2023. Vol. 127. P. 2151–2180. DOI: 10.1007/s00170-023-11517-8.
2. Antons O., Arlinghaus J. Distributing decision-making authority in manufacturing — review and roadmap for the factory of the future. *International Journal of Production Research*. 2022. Vol. 60, No. 13. DOI: 10.1080/00207543.2022.2057255.
3. Suvarna M., Yap K. S., Yang W., Li J., Ng Y. T., Wang X. Cyber-Physical Production Systems for Data-Driven, Decentralized, and Secure Manufacturing—A Perspective. *Engineering*. 2021. Vol. 7, No. 9. P. 1212–1223. DOI: 10.1016/j.eng.2021.04.021.
4. Okpoti E. S., Jeong I.-J. A reactive decentralized coordination algorithm for event-driven production planning and control: A cyber-physical production system prototype case study. *Journal of Manufacturing Systems*. 2021. Vol. 58. P. 143–158. DOI: 10.1016/j.jmsy.2020.11.002.
5. Zheng P., Xia L., Li C., Li X., Liu B. Towards Self-X cognitive manufacturing network: An industrial knowledge graph-based multi-agent reinforcement learning approach. *Journal of Manufacturing Systems*. 2021. Vol. 61. P. 16–26. DOI: 10.1016/j.jmsy.2021.08.002.

6. Paape N., van Eekelen J. A. W. M., Reniers M. A. Review of simulation software for cyber-physical production systems with intelligent distributed production control. *International Journal of Computer Integrated Manufacturing*. 2024. Vol. 37, No. 5. P. 589–611. DOI: 10.1080/0951192X.2023.2228270.

7. Antons O., Arlinghaus J. C. Data-driven and autonomous manufacturing control in cyber-physical production systems. *Computers in Industry*. 2022. Vol. 141. Article 103711. DOI: 10.1016/j.compind.2022.103711.

8. Helu M., Sobel W., Nelaturi S., Waddell R., Hibbard S. Industry Review of Distributed Production in Discrete Manufacturing. *Journal of Manufacturing Science and Engineering*. 2020. DOI: 10.1115/1.4046988.

ЛЕКЦІЯ 15. МЕТОДОЛОГІЯ ПРОЄКТУВАННЯ КІБЕРФІЗИЧНИХ ВИРОБНИЧИХ СИСТЕМ

Мета лекції

Мета лекції полягає у формуванні цілісної методології проєктування кіберфізичних виробничих систем — від аналізу технологічного об'єкта та визначення вимог до вибору архітектури, сенсорів, контролерів, виконавчих пристроїв, промислової мережі й програмних компонентів. Розглядаються функціональна декомпозиція, розподіл функцій між фізичними та цифровими компонентами, вибір технічних засобів, формування інформаційної та програмної архітектури, використання модельно-орієнтованого проєктування, а також перевірка взаємної узгодженості прийнятих рішень.

Проєктування КФВС відрізняється від проєктування звичайної локальної системи автоматизації тим, що необхідно одночасно узгодити фізичний процес, автоматичне керування, програмне забезпечення, комунікації, інформаційні моделі, аналітичні функції та кібербезпеку. Рішення, прийняте для одного рівня, може створити обмеження на іншому. Наприклад, перенесення алгоритму на віддалений сервер може спростити програмне супроводження, але збільшити затримку; додавання великої кількості сенсорів підвищує спостережуваність, але збільшує потік даних і кількість точок відмови.

Тому КФВС доцільно проєктувати як єдину систему, а не як послідовне додавання цифрових компонентів до вже сформованої автоматизації. Сучасні методології проєктування CPS також спираються на послідовний перехід від потреб і специфікацій до функціональної структури, моделей і реалізації [2, 3].

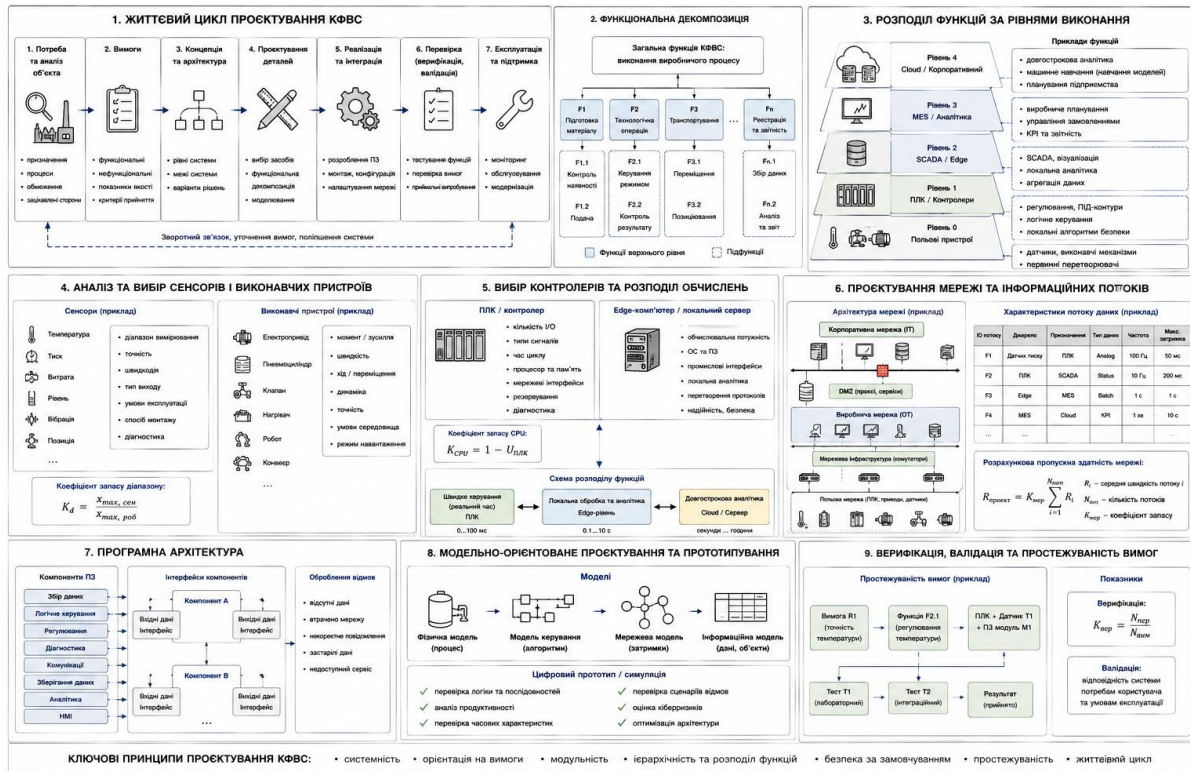


Рисунок 15.1 – Методологія проєктування кіберфізичних виробничих систем

15.1. Життєвий цикл проєктування КФВС

КФВС проходить не тільки стадію розроблення. Проєктне рішення повинно враховувати весь життєвий цикл: створення концепції, розроблення, виготовлення, введення в експлуатацію, використання, технічну підтримку, модернізацію та виведення з експлуатації.

ISO/IEC/IEEE 15288:2023 задає спільну систему процесів життєвого циклу для технічних систем і передбачає їх застосування до систем, їх елементів та систем систем протягом усього життєвого циклу [1].

Для КФВС це має практичний зміст. Якщо система проєктується лише для початкового запуску, подальша заміна сенсора, ПЛК або мережевого пристрою може потребувати значної переробки програмного забезпечення.

Раціональний процес проєктування можна представити послідовністю:

потреба → вимоги → концепція → функціональна структура → архітектура → технічна реалізація → інтеграція → перевірка → експлуатація → модернізація.

Ця послідовність не є строго односпрямованою. Результат вибору обладнання може вимагати повернення до архітектури, а моделювання — уточнення початкових вимог.

Кількість ітерацій проектування можна характеризувати:

$$N_{\text{ит}} = N_{\text{пов}} + 1,$$

де $N_{\text{ит}}$ — загальна кількість проходів проектного циклу; $N_{\text{пов}}$ — кількість повернень до попередніх етапів.

Велика кількість ітерацій сама по собі не означає помилкового процесу. Для складної КФВС уточнення вимог після моделювання або прототипування є природним. Проблемою є неконтрольована зміна вимог без простежуваного впливу на інші частини проекту.

Тому кожне суттєве проектне рішення повинно бути пов'язане з вимогою, яку воно реалізує.

15.2. Аналіз технологічного об'єкта та формування вимог

Проектування необхідно починати не з вибору ПЛК, мережі або програмної платформи, а з технологічного об'єкта.

На першому етапі встановлюють:

- призначення виробничої системи;
- технологічні операції;
- матеріальні потоки;
- режими роботи;
- змінні стану;
- керувальні впливи;
- зовнішні збурення;
- вимоги до продуктивності;
- аварійні режими;
- взаємодію з оператором.

Після цього визначаються функціональні та нефункціональні вимоги.

Функціональна вимога описує, що повинна виконувати система. Наприклад: підтримувати температуру, переміщувати виріб, реєструвати параметри, визначати стан обладнання.

Нефункціональна вимога визначає умови виконання: точність, допустимий час реакції, доступність, рівень захисту, обсяг архіву або можливість реконфігурації.

Для вимірюваного параметра можна задати допустиму відносну похибку:

$$\delta_{\text{доп}} = \frac{|\Delta x_{\text{доп}}|}{|x_{\text{ном}}|} \cdot 100\%,$$

де $\delta_{\text{доп}}$ — допустима відносна похибка; $\Delta x_{\text{доп}}$ — допустима абсолютна похибка; $x_{\text{ном}}$ — номінальне або характерне значення параметра.

Аналогічно для часової функції задається граничний час реакції, для мережі — допустима затримка, для архіву — потрібний період зберігання.

Вимога повинна бути перевірюваною. Формулювання «система повинна швидко реагувати» не дозволяє визначити, чи виконаний проєкт. Натомість вимога «команда повинна бути сформована не пізніше заданого граничного часу» має однозначний критерій перевірки.

На цьому ж етапі визначають **межі системи**: які компоненти належать до КФВС, а які є зовнішнім середовищем. Без чіткої межі неможливо однозначно встановити відповідальність за інтерфейси, дані та функції.

15.3. Функціональна декомпозиція

Після визначення вимог загальну функцію системи поділяють на підфункції.

Наприклад, для автоматизованої виробничої комірки можна виділити:

- визначення наявності виробу;
- ідентифікацію виробу;
- позиціювання;
- виконання технологічної операції;
- контроль результату;
- транспортування;
- реєстрацію параметрів;
- діагностування;
- обмін із MES.

Функції спочатку визначають незалежно від конкретного технічного засобу. Не потрібно відразу вирішувати, що певний алгоритм виконує конкретний ПЛК або edge-комп'ютер.

Таке відокремлення дозволяє пізніше раціонально розподілити функції між компонентами.

Функціональну структуру можна подати:

$$\mathcal{F}_{\Sigma} = \{ \mathcal{F}_1, \mathcal{F}_2, \dots, \mathcal{F}_{N_F} \},$$

де $\mathcal{F}_{\Sigma}$ — множина функцій КФВС; N_F — кількість виділених функцій.

Для кожної функції потрібно встановити:

- вхідні дані;
- результат;
- допустимий час виконання;
- необхідні ресурси;
- залежність від інших функцій;

- поведінку при відмові.

Не всі функції повинні мати однаковий рівень деталізації. Регулювання швидкості привода може вимагати опису мілісекундного циклу, а формування добового виробничого звіту — ні.

Функціональна декомпозиція дозволяє побачити, які задачі критичні до часу і повинні залишатися локальними, а які можна переносити на edge-, server- або cloud-рівень.

Саме перехід від специфікацій до формальної моделі функцій використовується у сучасних model-based підходах до проєктування CPS [3].

15.4. Формування архітектури КФВС

На архітектурному етапі визначається, **де саме виконуватиметься кожна функція** і як компоненти будуть взаємодіяти.

Типова багаторівнева структура може містити:

фізичний процес → сенсори та виконавчі пристрої → ПЛК → edge-рівень → SCADA/MES → корпоративні або cloud-сервіси.

Функцію доцільно розміщувати якомога ближче до фізичного об'єкта, якщо вона:

- критична до часу;
- потрібна для безпечної роботи;
- повинна виконуватися без зовнішньої мережі;
- використовує високочастотні локальні дані.

Навпаки, серверний або cloud-рівень доцільний для довготривалої аналітики, порівняння великих масивів і ресурсомісткого навчання моделей.

Для j -го варіанта розміщення функції можна сформулювати критерій:

$$J_j = w_t T_j + w_r R_j + w_c C_j,$$

де J_j — інтегральна оцінка варіанта; w_t — вага часової складової; T_j — очікуваний час отримання результату; w_r — вага ресурсної складової; R_j — потреба в обчислювальних або мережевих ресурсах; w_c — вага вартості; C_j — оцінка витрат реалізації.

Такий критерій придатний лише для варіантів, які вже виконують обов'язкові вимоги. Якщо архітектура не забезпечує граничний час аварійної реакції, вона не повинна ставати прийнятною через низьку вартість.

Під час формування архітектури також визначають:

- централізацію або децентралізацію;

- резервування;
- межі мережевих сегментів;
- джерела часу;
- власників даних;
- точки інтеграції;
- поведінку при втраті зв'язку.

РАМІ 4.0 може використовуватися як референсна структура для систематизації функцій, рівнів і життєвого циклу виробничих активів, але не є готовою схемою конкретної КФВС [4, 5].

15.5. Вибір сенсорної та виконавчої підсистем

Вибір сенсорів виконується після встановлення того, **який фізичний стан необхідно спостерігати.**

Для кожного вимірювального каналу аналізують:

- фізичну величину;
- діапазон;
- потрібну точність;
- швидкодію;
- частоту вимірювання;
- умови середовища;
- спосіб монтажу;
- тип вихідного сигналу;
- можливість діагностування.

Необхідно уникати ситуації, коли датчик вибрано за високою паспортною точністю, але його динамічні характеристики не відповідають процесу.

Для сенсорного каналу запас за діапазоном можна оцінити:

$$K_d = \frac{x_{\max, \text{сен}}}{x_{\max, \text{роб}}},$$

де K_d — коефіцієнт запасу вимірювального діапазону; $x_{\max, \text{сен}}$ — верхня межа датчика; $x_{\max, \text{роб}}$ — максимальне очікуване робоче значення.

Надмірний запас також небажаний: використання невеликої частини діапазону може погіршити фактичну роздільну здатність вимірювального тракту.

Виконавчі пристрої вибираються за:

- потрібним моментом або зусиллям;
- швидкістю;
- переміщенням;

- динамікою;
- точністю;
- режимом навантаження;
- умовами середовища.

Потужність або зусилля не є єдиними критеріями. Якщо привод задовольняє статичне навантаження, але не забезпечує потрібного прискорення, весь технологічний цикл може не виконувати вимогу продуктивності.

Сенсор і виконавчий механізм потрібно аналізувати разом з об'єктом і контуром керування, а не як незалежні компоненти.

15.6. Вибір контролерів та розподіл обчислень

Після визначення функцій оцінюється необхідна обчислювальна платформа.

Для ПЛК істотними є:

- кількість входів-виходів;
- типи сигналів;
- час виконання циклу;
- обсяг пам'яті;
- підтримка потрібних мереж;
- можливість регулювання;
- засоби діагностування;
- резервування;
- умови експлуатації.

Резерв обчислювального ресурсу:

$$K_{\text{CPU}} = 1 - U_{\text{ПЛК}},$$

де K_{CPU} — відносний вільний ресурс процесора.

Робота контролера постійно біля граничного завантаження небажана. Після введення в експлуатацію часто додаються діагностичні функції, мережевий обмін або нові сигнали.

Однак не потрібно автоматично вибирати максимально продуктивний контролер. Це може збільшити вартість без функціональної користі.

При складній аналітиці до ПЛК додається edge-вузол. Йому можна передати:

- спектральний аналіз;
- комп'ютерний зір;
- локальне ML-прогнозування;
- перетворення протоколів;

- агрегацію великих потоків.

Межа між ПЛК і edge-рівнем визначається вимогою реального часу. Втрата edge-комп'ютера не повинна порушувати базове безпечне локальне керування, якщо інше не передбачено спеціальною відмовостійкою архітектурою.

15.7. Проєктування комунікаційної та інформаційної структури

Після визначення взаємодіючих компонентів формується комунікаційна структура.

Для кожного інформаційного потоку потрібно встановити:

- джерело;
- одержувача;
- обсяг повідомлення;
- частоту передавання;
- допустиму затримку;
- допустимий джиттер;
- вимоги до синхронізації;
- поведінку при втраті.

Розрахункова сумарна потреба у пропускній здатності:

$$R_{\text{проект}} = K_{\text{мер}} \sum_{i=1}^{N_{\text{пот}}} R_i,$$

де $R_{\text{проект}}$ — потрібна розрахункова пропускна здатність; $K_{\text{мер}}$ — коефіцієнт запасу мережевої продуктивності; $N_{\text{пот}}$ — кількість інформаційних потоків.

Пропускна здатність, однак, не визначає часову детермінованість. Для керування рухом потрібна мережа з відповідними гарантіями часу, тоді як для архіву може бути достатньо звичайного асинхронного обміну.

Далі формується **інформаційна структура**. Для кожного параметра потрібно встановити:

- найменування;
- фізичний зміст;
- одиницю;
- тип;
- ідентифікатор об'єкта;
- часову мітку;
- джерело істини.

Це запобігає ситуації, коли декілька систем зберігають параметр під різними назвами або незалежно змінюють його значення.

Для цифрових двійників інформаційний обмін також повинен бути частиною архітектури. ISO 23247-1:2021 задає загальні принципи manufacturing digital twin framework, а ISO 23247-4:2021 встановлює вимоги до інформаційного обміну між його сутностями [6, 7].

15.8. Проєктування програмної архітектури

Програмне забезпечення КФВС не повинно утворювати один великий блок, у якому технологічна логіка, комунікації, аналітика та візуалізація взаємно залежать одна від одної.

Доцільно виділяти окремі програмні компоненти:

- збирання даних;
- логічне керування;
- регулювання;
- керування станами;
- діагностування;
- комунікації;
- зберігання;
- аналітику;
- НМІ.

Для програмного модуля повинен бути визначений інтерфейс:

вхідні дані → функція → вихідні дані → стан помилки.

Це дозволяє окремо тестувати компоненти та змінювати внутрішню реалізацію без перебудови всієї системи.

Model-Based Engineering використовує формальні моделі як основний засіб узгодження структури, поведінки та керування. Parant та співавтори запропонували CPS-методологію, де SysML використовується для опису системи, а керувальні застосунки пов'язуються з компонентною моделлю автоматизації [3].

Програмна архітектура повинна також передбачати оброблення відмов. Для кожної зовнішньої залежності необхідно визначити поведінку, якщо:

- немає даних;
- втрачено мережу;
- повідомлення некоректне;
- зовнішній сервіс недоступний;
- значення застаріло.

Наприклад, ПЛК не повинен нескінченно використовувати останню уставку з MES, якщо її актуальність обмежена.

Кожна важлива змінна тому повинна мати не тільки значення, а й інформацію про її валідність та час формування.

15.9. Модельно-орієнтоване проєктування та цифровий прототип

У складній КФВС фізичний прототип усієї системи є дорогим способом пошуку архітектурних помилок. Значну частину рішень доцільно перевіряти на моделях.

Модельно-орієнтоване проєктування може охоплювати:

- структуру системи;
- фізичну динаміку;
- стани;
- алгоритми;
- інформаційні потоки;
- мережеві затримки.

Samrao та Peças запропонували CPSD² як методологію, що охоплює процес створення, проєктування, побудови та впровадження CPS, починаючи з виробничих і бізнес-потреб [2].

Сучасні дослідження manufacturing system design також переходять до поєднання системного мислення, формального моделювання, процесів проєктування й цифрових засобів підтримки [8].

На основі моделей можна створити цифровий прототип і до фізичного монтажу перевірити:

- логічні послідовності;
- продуктивність;
- конфлікти ресурсів;
- час реакції;
- обсяг мережевого трафіку;
- поведінку при відмові.

Проте модель не усуває потреби у фізичній перевірці. Вона дозволяє раніше виявити частину помилок.

Якщо похибка моделі перевищує допустимий рівень:

$$E_{\text{мод}} > E_{\text{доп}},$$

де $E_{\text{доп}}$ — допустима для конкретної проєктної задачі похибка моделі.

Модель не повинна використовуватися для обґрунтування відповідного рішення без уточнення.

15.10. Верифікація, валідація та простежуваність вимог

Фінальна перевірка КФВС складається щонайменше з двох різних задач.

Верифікація перевіряє, чи реалізовано систему відповідно до її специфікації.

Валідація перевіряє, чи відповідає створена система реальній потребі та умовам застосування.

Це принципово різні питання. Система може безпомилково реалізувати неправильно сформульовану вимогу.

Для набору вимог можна визначити показник перевірності:

$$K_{\text{вер}} = \frac{N_{\text{пер}}}{N_{\text{вим}}},$$

де $K_{\text{вер}}$ — частка вимог, для яких отриманий результат перевірки; $N_{\text{пер}}$ — кількість перевірених вимог; $N_{\text{вим}}$ — загальна кількість вимог.

Проте навіть $K_{\text{вер}}=1$ не означає, що система правильна, якщо частина вимог була сформована неповно або суперечливо.

Простежуваність повинна забезпечувати ланцюг:

потреба → вимога → функція → компонент → реалізація → тест → результат.

Це дозволяє оцінити наслідок зміни. Якщо змінюється вимога до часу реакції, можна визначити, які функції, мережеві канали, контролери та випробування потребують перегляду.

До верифікації КФВС належать:

- перевірка програмної логіки;
- тестування інтерфейсів;
- вимірювання часових характеристик;
- перевірка відмов;
- тестування мережі;
- перевірка інформаційних моделей;
- кібербезпекове тестування.

Кібербезпека повинна закладатися в архітектуру, а не додаватися після завершення системи. ІЕС 62443-3-2:2020 передбачає оцінювання ризику саме на стадії системного проектування та формування zones, conduits і цільових вимог захисту [9].

Тільки після узгодженої перевірки фізичної, програмної, комунікаційної та інформаційної частин КФВС можна розглядати як завершений системний проєкт.

Висновки до лекції 15

Проектування кіберфізичної виробничої системи повинно починатися з аналізу технологічного процесу та формування

перевірюваних вимог. Вибір конкретного контролера, мережі або програмної платформи є наступним, а не початковим етапом.

Функціональна декомпозиція дозволяє відокремити завдання системи від способу їх реалізації. Після цього функції розподіляються між локальними контролерами, edge-, серверними та іншими рівнями відповідно до часових, ресурсних і безпекових обмежень.

Архітектура визначає не тільки склад обладнання, а й місце виконання алгоритмів, структуру інформаційних потоків, поведінку при відмовах, резервування та межі відповідальності компонентів.

Сенсори й виконавчі пристрої вибираються з урахуванням фізики процесу. Для сенсора мають значення не лише діапазон і точність, а й динаміка та умови монтажу. Для виконавчого пристрою разом із потужністю необхідно перевіряти швидкодію, точність та обмеження.

ПЛК доцільно використовувати для детермінованого локального керування, тоді як ресурсомістку аналітику можна переносити на edge- або серверний рівень. Такий розподіл не повинен створювати залежність критичних функцій від нестабільних зовнішніх сервісів.

Комунікаційна мережа проєктується за характеристиками конкретних інформаційних потоків. Номінальна пропускна здатність не замінює аналізу затримки, джиттера, синхронізації та поведінки при втраті повідомлень.

Програмна архітектура повинна бути модульною та мати чітко визначені інтерфейси. Модельно-орієнтований підхід дозволяє узгодити різні частини КФВС та перевірити значну кількість рішень ще до фізичного запуску.

Завершення проєктування потребує верифікації, валідації та простежуваності вимог. Кожна суттєва функція повинна мати зв'язок із початковою потребою, конкретною реалізацією та способом перевірки.

Контрольні питання

1. Чому проєктування КФВС не слід починати з вибору ПЛК?
2. Які стадії охоплює життєвий цикл технічної системи?
3. Чим функціональна вимога відрізняється від нефункціональної?
4. Чому вимога повинна бути кількісно або однозначно перевірюваною?
5. Що розуміють під функціональною декомпозицією?
6. Які характеристики необхідно задавати для кожної функції КФВС?
7. За якими критеріями визначається рівень виконання функції?
8. Які функції доцільно залишати на ПЛК?
9. Які задачі можуть виконуватися на edge-рівні?

10. За якими характеристиками вибирається промисловий сенсор?
11. Чому надмірний діапазон датчика не завжди є перевагою?
12. Які параметри необхідно враховувати при виборі виконавчого пристрою?
13. За якими показниками оцінюється контролер?
14. Які характеристики інформаційного потоку необхідно визначити при проєктуванні мережі?
15. Що розуміють під джерелом істини для виробничого параметра?
16. Чому програмну архітектуру КФВС доцільно будувати модульно?
17. Яке призначення модельно-орієнтованого проєктування?
18. Чим верифікація відрізняється від валідації?
19. Що розуміють під простежуваністю вимог?
20. Чому кібербезпеку необхідно враховувати вже на архітектурній стадії?

Список використаних джерел

1. ISO/IEC/IEEE 15288:2023. Systems and software engineering — System life cycle processes. 2nd ed. Geneva : International Organization for Standardization, 2023. 116 p.
2. Sampayo M., Peças P. CPSD²: A new approach for cyber-physical systems design and development. *Journal of Industrial Information Integration*. 2022. Vol. 28. Article 100348. DOI: 10.1016/j.jii.2022.100348.
3. Parant A., Gellot F., Zander D., Carré-Ménétrier V., Philippot A. Model-based engineering for designing cyber-physical systems from product specifications. *Computers in Industry*. 2023. Vol. 145. Article 103808. DOI: 10.1016/j.compind.2022.103808.
4. Schichtel M., Brossog M., Kampker A. et al. Towards a domain-specific information architecture enabling the investigation and optimization of flexible production systems by utilizing artificial intelligence. *The International Journal of Advanced Manufacturing Technology*. 2022. DOI: 10.1007/s00170-022-10141-2.
5. Javaid M., Haleem A., Singh R. P., Suman R. A primer on the factories of the future. *Sensors*. 2022. Vol. 22, No. 15. Article 5834. DOI: 10.3390/s22155834.
6. ISO 23247-1:2021. Automation systems and integration — Digital twin framework for manufacturing — Part 1: Overview and general principles. Geneva : International Organization for Standardization, 2021.
7. ISO 23247-4:2021. Automation systems and integration — Digital twin framework for manufacturing — Part 4: Information exchange. Geneva : International Organization for Standardization, 2021.
8. Leng J., Guo J., Xie J. et al. Review of manufacturing system design in the interplay of Industry 4.0 and Industry 5.0 (Part I): Design thinking and modeling methods. *Journal of Manufacturing Systems*. 2024. Vol. 76. P. 158–187. DOI: 10.1016/j.jmsy.2024.07.012.

9. IEC 62443-3-2:2020. Security for industrial automation and control systems — Part 3-2: Security risk assessment for system design. Geneva : International Electrotechnical Commission, 2020.

ЛЕКЦІЯ 16. КІБЕРФІЗИЧНІ ВИРОБНИЧІ СИСТЕМИ В КОНЦЕПЦІЇ INDUSTRY 5.0

Мета лекції

Мета лекції полягає у формуванні цілісного уявлення про подальший розвиток кіберфізичних виробничих систем у межах концепції Industry 5.0. Розглядаються людиноцентричність, взаємодія людини з автоматизованими та роботизованими системами, стале використання ресурсів, стійкість виробництва до збурень, застосування штучного інтелекту і цифрових двійників, а також принципи багатокритеріального оцінювання КФВС.

Industry 5.0 не означає відмову від технологій Industry 4.0. Кіберфізичні системи, IoT, промислові мережі, цифрові двійники, машинне навчання, хмарні та периферійні обчислення залишаються технологічною основою виробництва. Змінюється критерій, за яким оцінюється доцільність їх використання. Європейська концепція Industry 5.0 формулює три взаємопов'язані орієнтири: **людиноцентричність, сталість і стійкість** [1]. Таким чином, цифровізація розглядається не як самостійна мета, а як засіб отримання виробничого результату за одночасного врахування потреб людини, ресурсних обмежень і здатності системи функціонувати при порушеннях [1, 2].

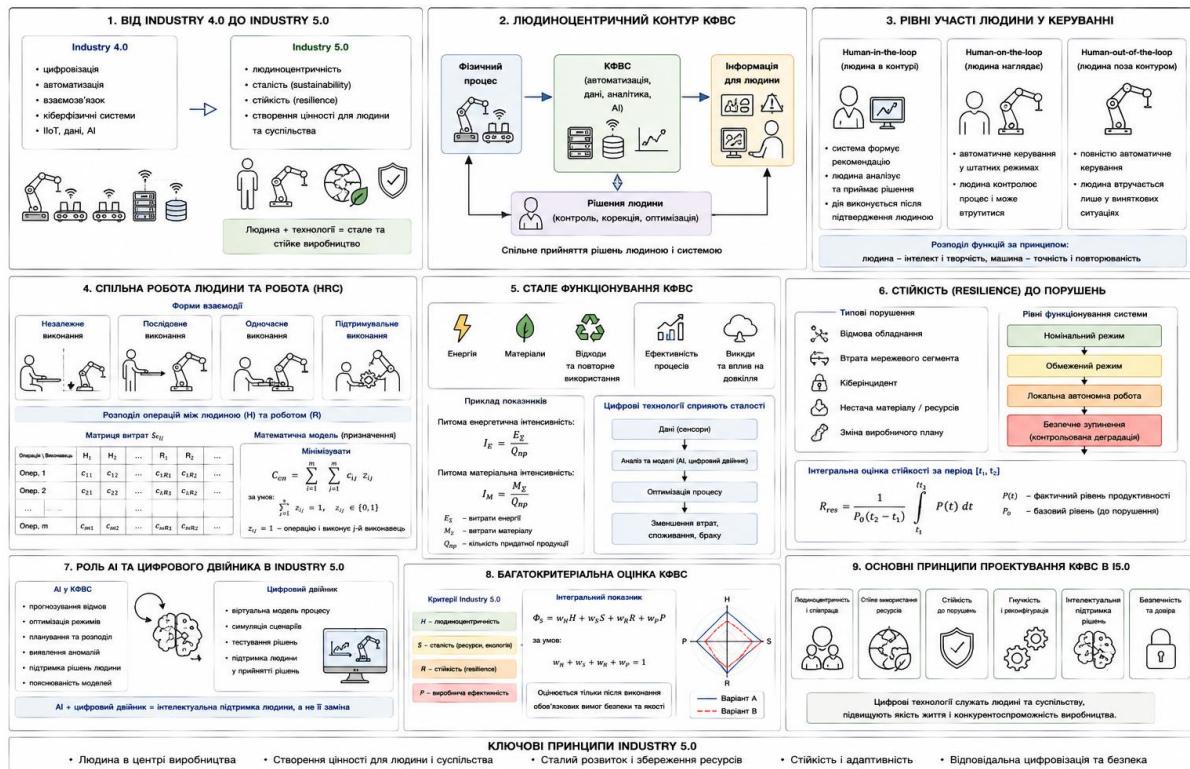


Рисунок 16.1 – Кіберфізичні виробничі системи в концепції industry 5.0

16.1. Перехід від Industry 4.0 до Industry 5.0

Industry 4.0 сформувала технологічну основу цифрового виробництва: кіберфізичні системи, мережеву інтеграцію, IoT, автономні компоненти, цифрові моделі та використання великих масивів виробничих даних.

Водночас високий рівень автоматизації сам по собі не гарантує раціональності виробничої системи. Система може мати високу продуктивність, але споживати надмірні ресурси, бути складною для оператора або виявитися нестійкою до порушення постачання чи відмови окремого цифрового сервісу.

Industry 5.0 розширює множину критеріїв, за якими оцінюється виробництво [1, 2]:

Industry 4.0: цифровізація, автоматизація, продуктивність, взаємозв'язок;

Industry 5.0: цифровізація + людиноцентричність + сталість + стійкість.

Такий перехід не є хронологічною заміною однієї «промислової революції» іншою. У сучасній науковій літературі Industry 5.0 переважно

розглядається як доповнення технологічної парадигми Industry 4.0 новою системою цілей і обмежень [2–4].

Для КФВС це означає, що під час проєктування вже недостатньо оцінювати тільки час циклу, точність регулювання або пропускну здатність мережі. Необхідно також з'ясувати:

- як система взаємодіє з оператором;
- які функції доцільно автоматизувати;
- які рішення повинні залишатися за людиною;
- як змінюється ресурсоспоживання;
- що відбудеться при відмові цифрових компонентів;
- наскільки швидко система може відновитися після порушення.

Таким чином, КФВС у Industry 5.0 проєктується не як максимально автономна система, а як система з обґрунтованим розподілом функцій між людиною, автоматикою та інтелектуальними цифровими компонентами.

16.2. Людиноцентрична кіберфізична виробнича система

Людиноцентричність не означає простого повернення оператора до операцій, які вже можуть виконуватися автоматично. Її зміст полягає у такому розподілі функцій, при якому технологія розширює можливості людини, а не змушує її пристосовуватися до непрозорої логіки автоматизованої системи.

У людиноцентричній КФВС людина може виконувати функції:

- постановки виробничої мети;
- вибору між альтернативними рішеннями;
- контролю нестандартних режимів;
- інтерпретації складних ситуацій;
- коригування автоматичного плану;
- підтвердження критичних дій;
- навчання та уточнення моделей.

Автоматизованій частині доцільно передавати повторювані обчислення, фізично важкі операції, високочастотне регулювання, роботу з великими масивами даних та функції, для яких машина має об'єктивні переваги.

Ли та співавтори пов'язують human-centric manufacturing із переходом від простого співіснування людини й автоматизованої машини до кооперації та глибшої спільної роботи, при якій враховуються фізичні й когнітивні потреби працівника [3].

Структурно людиноцентричний контур можна представити:

фізичний процес → КФВС → інформація для людини → рішення людини → КФВС → фізичний процес.

Поряд із ним залишаються автоматичні замкнені контури, де участь людини недоцільна через малий час реакції.

Тому людиноцентричність не означає, що оператор повинен підтверджувати кожну дію ПЛК. Навпаки, невдалий розподіл відповідальності може збільшити когнітивне навантаження та кількість помилок.

Основна інженерна задача полягає у визначенні **рівня автоматизації**, який відповідає конкретній функції.

16.3. Людина в контурі прийняття рішень

Для інтелектуальної КФВС доцільно розрізняти кілька варіантів взаємодії.

Human-in-the-loop

Людина є необхідною частиною циклу. Система формує інформацію або рекомендацію, але дія виконується лише після її рішення.

Наприклад:

модель прогнозує погіршення стану → інженер аналізує причину → приймає рішення про обслуговування.

Human-on-the-loop

Основне керування виконується автоматично, а людина контролює його результат і може втрутитися.

Цей підхід характерний для автоматизованого планування та частини роботизованих операцій.

Human-out-of-the-loop

Рішення і дія виконуються автоматично. Такий режим допустимий для функцій, поведінка яких достатньо формалізована, перевірена та обмежена технічними засобами.

Не можна встановити один рівень автономності для всієї КФВС. Швидкий ПД-контур може працювати повністю автоматично, тоді як зміна принципово нового технологічного режиму може потребувати рішення технолога.

При проектуванні необхідно визначити не тільки **що автоматизувати**, а й:

- хто має право скасувати автоматичне рішення;
- яку інформацію він при цьому бачить;
- за який час може втрутитися;
- якою буде поведінка при відсутності відповіді.

Людиноцентричне smart manufacturing у сучасних дослідженнях саме тому охоплює не лише інтерфейси HMI, а сенсорне сприйняття стану людини й обладнання, оброблення даних, передавання інформації та організацію власне взаємодії [4].

16.4. Спільна робота людини та робота

Одним із характерних прикладів Industry 5.0 є human–robot collaboration.

Традиційний промисловий робот часто працює у фізично відокремленій зоні. У колаборативній системі людина і робот можуть виконувати взаємопов'язані задачі в одному виробничому середовищі.

Сучасні огляди виділяють різні форми спільної роботи — незалежне, послідовне, одночасне й безпосередньо підтримувальне виконання операцій [5]. При цьому продуктивність не є єдиним критерієм: необхідно враховувати безпеку, гнучкість та ергономіку [5].

Задачу розподілу операцій між людиною та роботом можна формалізувати як задачу призначення.

$$\min C_{\text{сп}} = \sum_{i=1}^{N_{\text{оп}}} \sum_{j=1}^{N_{\text{вик}}} c_{ij} z_{ij},$$

де $C_{\text{сп}}$ — сумарний критерій призначення операцій; $N_{\text{оп}}$ — кількість операцій; $N_{\text{вик}}$ — кількість можливих виконавців; c_{ij} — оцінка виконання i -ї операції j -м виконавцем; z_{ij} — бінарна змінна призначення.

Для кожної операції:

$$\sum_{j=1}^{N_{\text{вик}}} z_{ij} = 1,$$

Вартісний коефіцієнт c_{ij} може враховувати тривалість, енергоспоживання, фізичне навантаження на оператора та інші фактори.

Проте не всі операції повинні входити до оптимізації як взаємозамінні. Частина з них через безпекові або технологічні вимоги може бути заздалегідь закріплена за конкретним типом виконавця.

Таким чином, математична оптимізація виконується тільки в межах допустимого простору співпраці.

16.5. Стале функціонування КФВС

Друга складова Industry 5.0 — sustainability — вимагає оцінювати виробництво не тільки за кількістю виготовленої продукції.

Для КФВС істотними стають:

- витрати електроенергії;
- використання сировини;

- втрати матеріалів;
- кількість браку;
- можливість повторного використання ресурсів;
- тривалість життєвого циклу обладнання.

Питомі витрати електроенергії можна визначити:

$$I_E = \frac{E_{\Sigma}}{Q_{\text{пр}}},$$

де I_E — енергетична інтенсивність виробництва; E_{Σ} — сумарні енергетичні витрати за досліджуваний період; $Q_{\text{пр}}$ — кількість придатної продукції.

Використання саме придатної продукції у знаменнику дозволяє врахувати, що енергія, витрачена на брак, не створила корисного виробничого результату.

Аналогічно матеріальну інтенсивність можна визначити:

$$I_M = \frac{M_{\Sigma}}{Q_{\text{пр}}},$$

де I_M — матеріальна інтенсивність; M_{Σ} — сумарні витрати відповідного матеріалу.

Цифрові технології можуть впливати на ці показники не безпосередньо, а через зміну виробничої поведінки.

Наприклад:

сенсорні дані → виявлення нестабільного режиму → корекція процесу → зменшення браку → зменшення питомих ресурсних витрат.

Тому екологічна або ресурсна користь цифровізації повинна підтверджуватися фактичними показниками. Сам факт встановлення додаткових сенсорів, серверів або цифрового двійника не є доказом сталості.

Industry 5.0 у сучасній літературі саме тому розглядається як поєднання цифрових технологій із цілями сталого створення цінності, а не як максимальне збільшення кількості цифрових компонентів [6].

16.6. Стійкість виробничої системи до порушень

Третій принцип Industry 5.0 — **resilience**, тобто здатність системи витримувати порушення, адаптуватися до них і відновлювати необхідний рівень функціонування.

Необхідно відрізняти стійкість у цьому розумінні від математичної стійкості системи автоматичного керування.

Для КФВС порушеннями можуть бути:

- відмова обладнання;
- відмова сенсора;
- втрата мережевого сегмента;
- недоступність cloud-сервісу;
- кіберінцидент;
- нестача виробничого ресурсу;
- раптова зміна виробничого плану.

Абсолютно нечутливою до всіх порушень систему зробити неможливо. Завдання полягає у збереженні критичних функцій і контрольованому відновленні.

Один із можливих нормованих показників виробничої стійкості за період порушення можна записати:

$$R_{\text{res}} = \frac{1}{P_0(t_2 - t_1)} \int_{t_1}^{t_2} P(t) dt,$$

де R_{res} — нормована інтегральна оцінка збереження продуктивності; $P(t)$ — фактичний рівень функціонування системи після порушення; P_0 — базовий рівень до порушення; t_1 — момент виникнення порушення; t_2 — момент завершення оцінюваного періоду.

При збереженні базового рівня протягом усього періоду значення такого показника дорівнює одиниці. Зниження продуктивності та тривале відновлення зменшують його.

Це один із можливих способів кількісної оцінки, а не універсальна стандартизована формула resilience.

Практичними засобами підвищення стійкості КФВС є:

- резервування;
- локальна автономність;
- реконфігурація;
- альтернативні виробничі маршрути;
- діагностування;
- резервне копіювання;
- контрольовані режими деградації.

Огляди resilient manufacturing підкреслюють, що складність сучасних взаємопов'язаних виробничих систем створює ризик каскадних порушень, через що особливого значення набувають локалізація відмов і можливість реконфігурації [8].

16.7. Контрольована деградація замість повної відмови

Для складної КФВС не кожна несправність повинна призводити до повного зупинення всього виробництва.

Можна передбачити декілька функціональних станів:

номінальний режим → обмежений режим → локальна автономна робота → безпечне зупинення.

Наприклад, при втраті cloud-сервісу локальні ПЛК повинні продовжити базове керування, навіть якщо тимчасово недоступна прогнозна аналітика.

При виході з ладу однієї виробничої машини багатоагентна система може перебудувати маршрут, якщо існує сумісний резервний ресурс.

При несправності допоміжного сенсора система може перейти до менш точного, але контрольованого режиму, якщо така поведінка була передбачена й перевірена.

Рівень збереженої функціональності можна оцінити:

$$K_{\text{функ}} = \frac{N_{\text{дост}}}{N_{\text{необх}}},$$

де $K_{\text{функ}}$ — частка доступних необхідних функцій; $N_{\text{дост}}$ — кількість функцій, які залишаються доступними; $N_{\text{необх}}$ — кількість функцій, необхідних для номінального режиму.

Для безпеки, однак, кількість функцій не є достатнім критерієм. Одна недоступна критична функція може вимагати негайного зупинення незалежно від високого $K_{\text{функ}}$.

Тому функції повинні попередньо класифікуватися за критичністю.

16.8. Штучний інтелект і цифрові двійники в Industry 5.0

AI в Industry 5.0 не повинен оцінюватися тільки за точністю прогнозу. Додаткового значення набувають:

- пояснюваність;
- надійність;
- контроль людиною;
- робота поза навчальним режимом;
- обмеження автономних дій.

Сучасний Industrial AI розглядається як один із технологічних засобів переходу до людиноцентричних, стійких і сталих виробничих систем, але його впровадження потребує поєднання алгоритмічної якості з реальними виробничими обмеженнями [7].

Приклад:

ML-модель → прогноз відмови 70 %

не є завершеним виробничим рішенням.

Потрібно додатково встановити:

- на яких даних отримано прогноз;
- чи перебуває обладнання в режимі, представленому під час навчання;
- які параметри найбільше вплинули на результат;
- яку дію можна виконати безпечно;
- чи має інженер можливість перевірити прогноз.

Цифровий двійник у людиноцентричній системі також змінює функціональний акцент. Він може бути не автономним «цифровим керівником» машини, а засобом підтримки прийняття рішень оператором, інженером або технологом. Дослідження 2024 року прямо розглядають human-centric digital twin як засіб поєднання обчислювальних можливостей цифрової моделі з критичним та творчим мисленням людини [9].

16.9. Багатокритеріальна оцінка КФВС

У попередніх поколіннях автоматизованих систем основними критеріями часто були продуктивність, точність і економічні витрати.

Для Industry 5.0 оцінка стає багатовимірною.

Один з можливих інтегральних критеріїв можна сформулювати:

$$\Phi_5 = w_{\text{люд}}H + w_{\text{стал}}S + w_{\text{стій}}R + w_{\text{вир}}P,$$

де Φ_5 — узагальнена оцінка варіанта КФВС; $w_{\text{люд}}$ — вага людиноцентричності; H — нормована оцінка людиноцентричних характеристик; $w_{\text{стал}}$ — вага сталості; S — нормована оцінка ресурсної та екологічної складової; $w_{\text{стій}}$ — вага стійкості; R — нормована оцінка resilience; $w_{\text{вир}}$ — вага виробничої ефективності; P — нормована виробнича оцінка.

Для ваг:

$$w_{\text{люд}} + w_{\text{стал}} + w_{\text{стій}} + w_{\text{вир}} = 1.$$

Такий показник є проєктним багатокритеріальним інструментом, а не стандартним показником Industry 5.0.

Окремі обов'язкові вимоги не можна компенсувати високими значеннями інших критеріїв. Небезпечна роботизована система не стає прийнятною через високу продуктивність. Так само висока енергоефективність не компенсує нездатність системи виконувати основну технологічну операцію.

Тому доцільна дворівнева оцінка:

- 1. перевірка обов'язкових обмежень;**
- 2. оптимізація серед допустимих варіантів.**

Саме цей принцип дозволяє уникнути хибної ситуації, коли інтегральний бал приховує критичний недолік.

16.10. Обмеження переходу до Industry 5.0

Industry 5.0 не може бути реалізована простим придбанням нового класу обладнання. Її принципи стосуються архітектури і способу організації виробництва.

До основних технічних обмежень належать:

- складність інтеграції обладнання різних поколінь;
- неповні або неякісні дані;
- відсутність єдиних інформаційних моделей;
- складність перевірки AI-рішень;
- кібербезпекові ризики;
- висока складність реконфігурованих систем.

Людиноцентричність також не можна звести до використання красивішого НМІ. Потрібно аналізувати реальний розподіл відповідальності, доступність інформації та можливість оператора зрозуміти стан системи.

Не вся автоматизація повинна бути автономною. У деяких випадках доцільніше залишити остаточне рішення за людиною, навіть якщо алгоритм технічно може сформувати його самостійно.

Не вся децентралізація підвищує стійкість. Збільшення кількості автономних компонентів одночасно збільшує складність їх координації.

Не вся цифровізація підвищує сталість. Сервери, сенсорна інфраструктура та обчислення також використовують енергію й матеріальні ресурси.

Отже, раціональний принцип Industry 5.0 можна сформулювати так: використовувати цифрову технологію там, де її внесок у виробничу, людську, ресурсну або стійкісну характеристику системи перевищує створені нею додаткові витрати та ризики.

Висновки до лекції 16

Industry 5.0 розвиває технологічну основу Industry 4.0, доповнюючи її трьома системними орієнтирами: людиноцентричністю, сталістю та стійкістю. Кіберфізичні системи при цьому залишаються основним механізмом інтеграції фізичних і цифрових виробничих процесів.

Людиноцентрична КФВС не усуває людину з виробництва і не повертає їй функції, які доцільніше виконуються автоматично. Завдання полягає у раціональному розподілі відповідальності між оператором, автоматичними контролерами та інтелектуальними цифровими компонентами.

Людина може перебувати безпосередньо в контурі прийняття рішень, виконувати функцію нагляду або не брати участі в конкретному швидкому контурі. Рівень автономності повинен визначатися окремо для кожної функції.

Спільна робота людини і робота дозволяє поєднати гнучкість та когнітивні можливості людини з точністю, повторюваністю та фізичними можливостями машини. Розподіл операцій при цьому повинен враховувати не тільки продуктивність, а безпеку й ергономіку.

Сталість КФВС потребує кількісного контролю питомих енергетичних і матеріальних витрат. Цифровізація є корисною лише тоді, коли вона фактично покращує ресурсні або виробничі показники, а не просто збільшує кількість цифрових компонентів.

Стійка КФВС повинна не тільки запобігати відмовам, а й зберігати критичні функції, працювати у контрольовано обмеженому режимі та відновлюватися після порушень.

Штучний інтелект, цифрові двійники, багатоагентні системи та edge/cloud-обчислення створюють технологічні можливості для Industry 5.0, але їх застосування має залишатися контрольованим, перевірюваним і пов'язаним із фізичними обмеженнями реального виробничого процесу.

Таким чином, завершальним критерієм якості КФВС стає не максимальний рівень автоматизації. Раціонально спроектована система повинна одночасно виконувати виробничу функцію, бути зрозумілою для людини, економно використовувати ресурси, витримувати порушення та залишатися керованою протягом життєвого циклу.

Контрольні питання

1. Які три основні орієнтири визначають концепцію Industry 5.0?
2. Чим Industry 5.0 відрізняється від Industry 4.0?
3. Чому Industry 5.0 не потрібно розглядати як відмову від технологій Industry 4.0?
4. Що розуміють під людиноцентричною КФВС?
5. Які функції доцільно залишати за людиною?
6. Чим Human-in-the-loop відрізняється від Human-on-the-loop?
7. Для яких функцій допустимий Human-out-of-the-loop?

8. Які форми взаємодії людини та колаборативного робота використовуються у виробництві?
9. За якими критеріями можна розподіляти операції між людиною та роботом?
10. Що розуміють під сталим функціонуванням виробничої системи?
11. Як визначається питома енергетична інтенсивність виробництва?
12. Чому сама цифровізація не є доказом сталості?
13. Що означає resilience виробничої системи?
14. Чим resilience відрізняється від математичної стійкості САК?
15. Які засоби дозволяють підвищити стійкість КФВС до порушень?
16. Що розуміють під контрольованою деградацією?
17. Яку роль можуть виконувати AI та цифрові двійники в людиноцентричній КФВС?
18. Чому точності ML-моделі недостатньо для прийняття відповідального виробничого рішення?
19. Для чого використовується багатокритеріальне оцінювання КФВС?
20. Чому обов'язкова безпекова вимога не повинна компенсуватися високою продуктивністю?

Список використаних джерел

1. Breque M., De Nul L., Petridis A. *Industry 5.0: Towards a Sustainable, Human-Centric and Resilient European Industry*. Luxembourg : Publications Office of the European Union, 2021.
2. Leng J., Sha W., Wang B., Zheng P., Zhuang C., Liu Q., Wuest T., Mourtzis D., Wang L. Industry 5.0: Prospect and retrospect. *Journal of Manufacturing Systems*. 2022. Vol. 65. P. 279–295. DOI: 10.1016/j.jmsy.2022.09.017.
3. Lu Y., Zheng H., Chand S., Xia W., Liu Z., Xu X., Wang L., Qin Z., Bao J. Outlook on human-centric manufacturing towards Industry 5.0. *Journal of Manufacturing Systems*. 2022. Vol. 62. P. 612–627. DOI: 10.1016/j.jmsy.2022.02.001.
4. Zhang C., Wang Z., Zhou G., Chang F., Ma D., Jing Y., Cheng W., Ding K., Zhao D. Towards new-generation human-centric smart manufacturing in Industry 5.0: A systematic review. *Advanced Engineering Informatics*. 2023. Vol. 57. Article 102121. DOI: 10.1016/j.aei.2023.102121.
5. Keshvarparast A., Battini D., Battaia O., Pirayesh A. Collaborative robots in manufacturing and assembly systems: literature review and future research agenda. *Journal of Intelligent Manufacturing*. 2024. Vol. 35. P. 2065–2118. DOI: 10.1007/s10845-023-02137-w.
6. van Erp T., Carvalho N. G. P., Gerolamo M. C., Gonçalves R., Rytter N. G. M., Gladysz B. Industry 5.0: A new strategy framework for sustainability management and beyond. *Journal of Cleaner Production*. 2024. Vol. 461. Article 142271. DOI: 10.1016/j.jclepro.2024.142271.

7. Leng J., Zhu X., Huang Z., Li X., Zheng P., Zhou X., Mourtzis D., Wang B., Qi Q., Shao H., Wan J., Chen X., Wang L., Liu Q. Unlocking the power of industrial artificial intelligence towards Industry 5.0: Insights, pathways, and challenges. *Journal of Manufacturing Systems*. 2024. Vol. 73. P. 349–363. DOI: 10.1016/j.jmsy.2024.02.010.
8. Strategies for Resilient Manufacturing: A Systematic Literature Review of Failure Management in Production. *Procedia CIRP*. 2024. Vol. 130. P. 1393–1402. DOI: 10.1016/j.procir.2024.10.257.
9. *Digital Twins for Industry 5.0: Unlocking the Human Potential*. *Procedia CIRP*. 2024. Vol. 130. P. 761–766. DOI: 10.1016/j.procir.2024.10.161

Електронне навчальне видання комбінованого використання
Можна використовувати в локальному мережному режимі

Мезеря Андрій Юрійович

ПРОЄКТУВАННЯ ТА ТЕХНОЛОГІЇ КІБЕРФІЗИЧНИХ ВИРОБНИЧИХ СИСТЕМ

Конспект лекцій
для здобувачів вищої освіти другого (магістерського) рівня
денної та заочної форм здобуття освіти за спеціальністю
G7 «Автоматизація, комп'ютерно-інтегровані технології та робототехніка»

В авторській редакції

Підписано до розміщення 23.10.2025. Гарнітура Times New Roman.
Ум. друк. арк. 6,49. Обсяг 6,389 Мб. Зам. № 498/25.

Харківський національний університет імені В. Н. Каразіна,
61022, м. Харків, майдан Свободи, 4.
Свідоцтво суб'єкта видавничої справи ДК № 3367 від 13.01.2009
Видавництво ХНУ імені В. Н. Каразіна