



Я, Губський Ігор Вячеславович, як здобувач вищої освіти ХНУРЕ, розумію та підтримую політику закладу з академічної доброчесності. Я не надавав і не одержував недозволену допомогу під час підготовки кваліфікаційної роботи. Я не використовував штучний інтелект для підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

«10» грудня 2025р.

  
\_\_\_\_\_

Ігор ГУБСЬКИЙ

# ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Факультет	Автоматики і комп'ютеризованих технологій
Кафедра	Комп'ютерно-інтегрованих технологій, автоматизації та робототехніки
Рівень вищої освіти	другий (магістерський)
Спеціальність	174 Автоматизація, комп'ютерно-інтегровані технології та робототехніка
Тип програми	освітньо-професійна
Освітня програма	Комп'ютеризовані та робототехнічні системи (код і повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри КІТАР \_\_\_\_\_  
(підпис)

«\_\_\_» \_\_\_\_\_ 2025 р.

## ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві \_\_\_\_\_ *Губському Ігору Вячеславовичу* \_\_\_\_\_  
(прізвище, ім'я, по батькові)

1. Тема роботи \_\_\_\_\_ *Розроблення системи автоматизації інформаційно-технічного забезпечення кіберфізичного виробництва* \_\_\_\_\_

затверджена наказом по університету від "10" листопада 2025р. № 1018 Ст.

2. Термін подання здобувачем роботи \_\_\_\_\_ "16" грудня 2025р. \_\_\_\_\_

3. Вихідні дані до роботи \_\_\_\_\_ *3.1 Сучасне кіберфізичне виробництво;* \_\_\_\_\_

*3.2 Кросплатформний стандарт промислової інтеграції – OPC UA;* \_\_\_\_\_

*3.3 Мова програмування – Python 3.11;* \_\_\_\_\_

*3.4 Середовище виконання JavaScript – Node.js;* \_\_\_\_\_

*3.5 База даних – PostgreSQL;* \_\_\_\_\_

*3.6 Протокол обміну повідомленнями – MQTT;* \_\_\_\_\_

*3.7 Програмна платформа – Docker;* \_\_\_\_\_

4. Перелік питань, що потрібно опрацювати в роботі \_\_\_\_\_ *4.1 Вступ;* \_\_\_\_\_

*4.2 Аналіз предметної області та існуючих рішень;* \_\_\_\_\_

*4.3 Розроблення концепції та архітектури автоматизованої системи;* \_\_\_\_\_

*4.4 Моделювання функціонування автоматизованої системи;* \_\_\_\_\_

*4.5 Розробка програмного забезпечення автоматизованої системи;* \_\_\_\_\_

*4.6 Висновки.* \_\_\_\_\_

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій Демонстраційний матеріал представлений у форматі презентації PowerPoint (\*.ppt) – 20 с. формату А4.

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

### КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	<i>Аналіз предметної області та існуючих рішень</i>	01.09 – 15.09.25	<i>виконано</i>
2	<i>Розроблення концепції та архітектури автоматизованої системи</i>	16.09 – 30.09.25	<i>виконано</i>
3	<i>Моделювання функціонування автоматизованої системи</i>	01.11 – 18.11.25	<i>виконано</i>
4	<i>Розробка програмного забезпечення автоматизованої системи</i>	19.11 – 02.12.25	<i>виконано</i>
5	<i>Оформлення пояснювальної записки</i>	03.12 – 05.12.25	<i>виконано</i>
6	<i>Подання роботи на нормоконтроль</i>	06.12 – 08.12.25	
7	<i>Подання роботи на перевірку Інтернет-системою StrikePlagiarism</i>	09.12 – 10.12.25	<i>виконано</i>
8	<i>Подання роботи на рецензію</i>	11.12 – 13.12.25	<i>виконано</i>
9	<i>Подання роботи на підпис зав. кафедри</i>	14.12 – 15.12.25	<i>виконано</i>
10	<i>Подання кваліфікаційної роботи в ЕК</i>	16.12.25	<i>виконано</i>

Дата видачі завдання 01.09.2025р.

Здобувач

  
(підпис)

Ігор ГУБСЬКИЙ

Керівник роботи

\_\_\_\_\_  
(підпис)

доцент Рауф АЛЛАХВЕРАНОВ  
(посада, власне ім'я прізвище)

## РЕФЕРАТ

Пояснювальна записка: 97 с., 2 табл., 17 рис., 2 дод., 37 джерел.

КІБЕРФІЗИЧНА СИСТЕМА, АВТОМАТИЗАЦІЯ. ІНФОРМАЦІЙНО-ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ, ВИРОБНИЦТВО, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, ПІД-РЕГУЛЯТОР, МОДЕЛЮВАННЯ, ІНДУСТРІЯ 4.0.

Мета роботи – підвищення ефективності функціонування кіберфізичного виробництва шляхом створення автоматизованої системи, яка забезпечує інтеграцію інформаційних потоків, моніторинг технічних процесів і підтримку прийняття рішень у реальному часі.

Об'єкт дослідження – процес інформаційно-технічного забезпечення кіберфізичного виробництва, що включає збір, передавання, оброблення та використання технологічних даних у режимі реального часу.

Предмет дослідження – методи, засоби та архітектурні рішення автоматизації інформаційно-технічного забезпечення кіберфізичних виробничих процесів.

У кваліфікаційній роботі розроблено автоматизовану систему інформаційно-технічного забезпечення кіберфізичного виробництва. Система забезпечує збір, передавання, аналітичну обробку та використання технологічних даних у режимі реального часу. Створено математичні моделі, алгоритми аналізу та ПІД-керування, а також програмне забезпечення на мові Python, що реалізує повний цикл функціонування кіберфізичної системи. Проведене моделювання підтвердило ефективність і стійкість запропонованих рішень.

Отримані результати роботи можна віднести до Цілі сталого розвитку 9 «Промисловість, інновації та інфраструктура», зокрема до пункту 9.4 «Розвиток високотехнологічного машинобудування».

## ABSTRACT

Explanatory note: 97 pp., 2 tab., 17 fig., 2 appendices, 37 sources.

CYBER-PHYSICAL SYSTEMS, AUTOMATION. INFORMATION AND TECHNICAL SUPPORT, MANUFACTURING, SOFTWARE, PID-REGULATOR, MODELING, INDUSTRY 4.0.

The aim of the work is to improve the efficiency of cyber-physical production by creating an automation system that integrates information flows, monitors technical processes, and supports real-time decision-making.

The object of research is cyber-physical production systems that combine physical processes, information technologies, and elements of artificial intelligence.

The object of research is the process of information and technical support for cyber-physical production, which includes the collection, transmission, processing, and use of technological data in real time.

The thesis develops a system for automating the information and technical support of cyber-physical production. The system provides real-time collection, transmission, analytical processing, and use of technological data. Mathematical models, analysis algorithms, and PID control have been created, as well as Python software that implements the full cycle of cyber-physical system functioning. The modeling confirmed the effectiveness and stability of the proposed solutions..

This work involved the development of a control system for a robot dog using intelligent mechatronic devices. In the course of the research, a set of theoretical and applied tasks was solved, which made it possible to achieve the set goal of the work.

The results of the work can be attributed to Sustainable Development Goal 9 “Industry, Innovation and Infrastructure,” in particular to paragraph 9.4 “Development of high-tech engineering”.

## ЗМІСТ

Перелік скорочень .....	9
Вступ .....	10
1 Аналіз предметної області та існуючих рішень .....	12
1.1 Сутність і структура кіберфізичних систем .....	12
1.2 Сучасні підходи до автоматизації виробничих процесів .....	14
1.3 Проблеми інформаційно-технічного забезпечення кіберфізичних систем .....	16
1.4 Формулювання вимог до системи .....	19
1.5 Висновки до першого розділу .....	21
2 Розроблення концепції та архітектури автоматизованої системи .....	23
2.1 Створення інтегрованої системи управління .....	23
2.2 Вибір принципів побудови .....	26
2.3 Розроблення загальної архітектури .....	28
2.4 Забезпечення інтеграції з зовнішніми системами .....	31
2.5 Інформаційна модель .....	35
2.6 Висновки до другого розділу .....	38
3 Моделювання функціонування автоматизованої системи .....	40
3.1 Моделювання роботи системи .....	40
3.2 Побудова математичних моделей .....	41
3.3 Результати тестування та аналіз ефективності системи .....	44
3.4 Порівняння системи з існуючими рішеннями .....	45
3.5 Висновки до третього розділу .....	46
4 Розробка програмного забезпечення автоматизованої системи .....	48
4.1 Загальна концепція алгоритмічного забезпечення .....	48
4.2 Структура алгоритмів управління системою .....	52
4.3 Архітектура програмного забезпечення .....	55

4.4 Вибір інструментальних засобів програмної реалізації .....	60
4.5 Програмна логіка взаємодії модулів .....	65
4.6 Діаграми алгоритмів та програмних процесів .....	69
4.7 Програмна реалізація системи .....	75
4.8 Тестування та оцінювання працездатності програмного забезпечення ..	83
4.9 Висновки до четвертого розділу .....	88
5 Безпечні умови праці у лабораторному приміщені .....	87
Висновки .....	92
Перелік джерел посилання .....	94
Додаток А Публікація за темою досліджень .....	98
Додаток Б Програмна реалізація головного модуля системи main.py .....	99
Додаток В Демонстраційний матеріал .....	100

## ПЕРЕЛІК СКОРОЧЕНЬ

- АЧХ – амплітудно-частотна характеристика;  
БД – база даних;  
КПО – коефіцієнт природної освітленості;  
КФС – кіберфізичні системи;  
П – пропорційний;  
ПЗ – програмне забезпечення;  
ПІ – пропорційно-інтегральний;  
ПІД – пропорційно-інтегрально-диференціальний;  
ПК – персональний комп'ютер;  
ПЛК – програмований логічний контролер;  
ТЗ – технічне завдання.

## ВСТУП

На сучасному етапі розвитку промисловості простежується перехід до нової парадигми – Індустрії 4.0, котра уособлює в собі цифровізацію, автоматизацію, а також інтеграцію фізичних і віртуальних процесів у межах єдиних виробничих комплексів. Одним із пріоритетних елементів даної парадигми можна назвати кіберфізичні системи (Cyber-Physical Systems, CPS), які реалізують взаємодію між обчислювальними засобами, фізичними об'єктами та комунікаційною інфраструктурою.

Утім упровадження більш складних виробничих процесів, посилення вимог щодо якості продукції, зниження витрат, а також підвищення рівня безпеки породжують створення інтелектуальних систем керування, що, зі свого боку, спроможні самостійно аналізувати дані й адаптувати параметри роботи в реальному часі, до того ж, прогнозувати відмови обладнання. У зв'язку з цим виникає затребуваність щодо використання ефективного інформаційно-технічного забезпечення, що втілює збір, оброблення, зберігання й передавання великих обсягів даних, і водночас реалізує взаємодію між різними рівнями виробничої системи.

Проте, промислові системи автоматизації, що функціонують наразі, здебільшого є фрагментованими, до того ж, характеризуються обмеженими можливостями інтеграції, а також не відповідають сучасним вимогам щодо швидкодії, масштабованості та безпеки. Це свідчить пронагальну потребу щодо розроблення нових підходів для побудови систем автоматизації, котрі будуть орієнтовані на принципи кіберфізичних систем, зокрема, мова йде про адаптивність, гнучкість, взаємозв'язок, а також залучення цифрових двійників.

Зважаючи на викладене вище, тема роботи є актуальною, адже розроблення системи автоматизації інформаційно-технічного забезпечення

кіберфізичного виробництва гарантуватиме оптимізацію виробничих процесів, підвищення ефективності керування, зниження ризику відмов обладнання та сприятиме конкуренто спроможності підприємств у цифровій економіці.

Мета роботи – підвищення ефективності функціонування кіберфізичного виробництва шляхом створення автоматизованої системи, яка забезпечує інтеграцію інформаційних потоків, моніторинг технічних процесів і підтримку прийняття рішень у реальному часі.

Об'єкт дослідження – процес інформаційно-технічного забезпечення кіберфізичного виробництва, що включає збір, передавання, оброблення та використання технологічних даних у режимі реального часу.

Предмет дослідження – методи, засоби та архітектурні рішення автоматизації інформаційно-технічного забезпечення кіберфізичних виробничих процесів.

Для досягнення поставленої мети доцільно розв'язати такі завдання:

- вивчити архітектури кіберфізичних систем, а також сформулювати вимоги до інформаційно-технічного забезпечення;
- розробити концептуальну модель автоматизованої системи для кіберфізичного виробництва.
- розробити алгоритми оброблення даних сенсорних мереж і керування технологічними процесами.
- утворити програмно-апаратний прототип системи.
- протестувати системи й оцінити ефективність автоматизації;
- провести заходи у лабораторії для забезпечення безбечних умов праці.

Кваліфікаційна робота виконана згідно методичних вказівок [1], положення про кваліфікаційну роботу [2], ДСТУ 3008-15 [3] та із використанням статей і матеріалів із наукових публікацій. Публікація за темою досліджень [4] наведена в Додатку А.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ІСНУЮЧИХ РІШЕНЬ

## 1.1 Сутність і структура кіберфізичних систем

Сучасні умови розвитку промисловості, що супроводжуються переходом до концепції Індустрії 4.0, провідну роль надають кіберфізичним системам (КФС), адже вони поєднують, інформаційні технології, засоби керування та фізичні процеси в суцільне інтегроване середовище, в якому можна виконувати моніторинг, аналізувати та керувати виробничими процесами в режимі реального часу [5, 6].

КФС подається як комплекс взаємопов'язаних програмно-апаратних засобів. Зі свого боку, вони забезпечують інтеграцію між фізичними об'єктами (сенсорами, роботами, виконавчими механізмами, машинами) та їхніми цифровими відтвореннями, зокрема, моделями, алгоритмами, базами даних (БД). Взаємодію між цими рівнями реалізовано за допомогою мережевих технологій, як-от промислового Інтернету речей (ІоТ)[6].

Серед характеристик КФС виокремимо головні [7]:

- інтелектуальне керування на основі аналізу даних;
- самоадаптація та самодіагностика системи;
- високий рівень автоматизації процесів прийняття рішень;
- інтеграція у глобальні інформаційні мережі;
- взаємодія в реальному часі між компонентами.

На рисунку 1.1 відображено типову структуру КФС.

Типова структура КФС ґрунтується на декількох функціональних рівнях:

- фізичному. Мова йде про сукупність технічних пристроїв, сенсорів, роботизованих систем, контролерів, технологічного обладнання, що безпосередньо впливають на фізичні процеси виробництва;

- комунікаційному. Він забезпечує обмін даними між пристроями, підсистемами та центральними вузлами керування, задіюючи промислові мережі (Ethernet/IP, Profinet, Modbus, OPCUA, MQTT тощо);
- кібернетичному (інформаційний). На цьому рівні виконується аналіз, оброблення та зберігання даних, тобто задіяні сервери, аналітичні модулі, БД, системи штучного інтелекту та цифрові двійники;
- керувальному. Він використовує алгоритми керування технологічними процесами, оптимізацію ресурсів і планування виробничих операцій, а також забезпечує зворотний зв'язок із фізичними об'єктами;
- користувацької взаємодії. Цей рівень надає людині інтерфейси для подальшого моніторингу, візуалізації та керування процесами (HMI, SCADA, MES-системи).

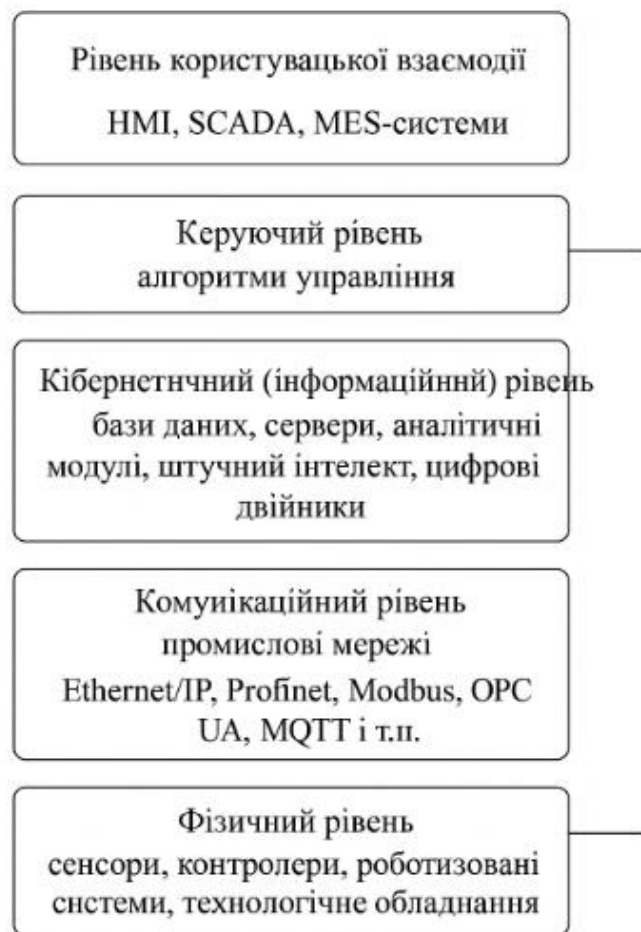


Рисунок 1.1 – Типова структура кіберфізичної системи

Шляхом інтеграції окреслених рівнів КФС створює замкнений контур керування, в якому інформація від сенсорів надходить до цифрового середовища, де аналізується алгоритмами, а вихідні результати, зі свого боку, впливають на фізичні процеси за допомогою виконавчих пристроїв. Описана взаємодія стає підґрунтям для інтелектуального виробництва, адже у такому форматі воно спроможне до оптимізації параметрів роботи, саморегулювання й адаптації до змін зовнішніх умов.

Однією з вагомих складових КФС вважається цифровий двійник, тобто віртуальна модель фізичного об'єкта чи процесу, котра відображає його стан у реальному часі. Застосування цифрових двійників дозволяє як проводити моделювання, так і прогнозувати несправності, отже, планувати технічне обслуговування без втручання до реального процесу [7-9].

З огляду, на сказане вище, КФС можна розглядати як основу для створення адаптивних, гнучких і разом з тим взаємопов'язаних виробничих комплексів. До того ж, їхнє впровадження сприяє підвищенню ефективності керування, зниженню витрат і водночас забезпеченню високої якості продукції, адже це пріоритетні цілі сучасного автоматизованого виробництва.

## 1.2 Сучасні підходи до автоматизації виробничих процесів

Автоматизація виробничих процесів набула значення одного із пріоритетних напрямків розвитку сучасного промислового виробництва, бо сприяє підвищенню ефективності використання ресурсів, стабільності якості продукції, зниженню впливу людського чинника, а також скороченню часу виробничого циклу. Так, автоматизовані системи еволюціонували від локальних контролерів і традиційних SCADA-систем до інтегрованих корпоративних платформ керування, котрими послуговуються у межах концепції Індустрії 4.0.

SCADA-системи були розроблені для збору, моніторингу із подальшим обробленням технологічних даних у реальному часі. Вони втілюють

диспетчерське керування процесами, а також відображення параметрів роботи обладнання із реєстрацією подій [8].

Головними функціями SCADA назвемо:

- збір інформації з сенсорів і контролерів;
- архівацію й аналіз даних;
- візуалізацію технологічних процесів;
- реалізацію дистанційного керування.

Власне типовими прикладами SCADA-систем є Wonderware, SiemensWinCC, Ignition, TRACEMODE.

MES-системи, зі свого боку, реалізують керування виробничими операціями на рівні цеху. Іншими словами, вони є проміжною ланкою між технологічними рівнями (SCADA, PLC) і системами планування ресурсів підприємства (ERP).

Ключовими завданнями MES-систем є:

- планування й диспетчеризація виробничих замовлень;
- відстеження якості продукції та формування звітності;
- контроль виконання операцій у реальному часі;
- керування персоналом і ресурсами.

Як приклади MES-рішень наведемо SiemensSIMATICIT, RockwellFactoryTalk, WonderwareMES.

ERP-системи, як правило, формують верхній рівень керування підприємством, тобто забезпечують комплексну координацію фінансових, матеріальних та інформаційних потоків. Зі свого боку, вони реалізують керування фінансами, логістикою, кадрами, запасами, а також охоплюють функції планування виробництва. Інтеграція ERP із нижчими рівнями автоматизації сприяє впровадженню прозорості узгодженості всіх бізнес-процесів. Найпопулярнішими системами цього класу є SAPERP, OracleERPcloud, MicrosoftDynamics 365 [10].

Підкреслимо, що розвиток Інтернету речей (IoT) та його промислової модифікації (IIoT) став рушійною силою для створення високорівневих систем

збору й аналізу даних, що надходять з незліченної кількості сенсорів і пристроїв. Так, IoT-платформи виконують підключення й ідентифікацію, обмін інформацією й аналітику з метою оптимізації виробничих процесів. Серед популярних рішень виокремимо: AWSIoT, SiemensMindSphere, PTCThingWorx, AzureIoTHub.

Задіюючи ПоТ, підприємства можуть утворювати розподілені системи моніторингу та керування, котрі інтегрують кіберфізичні компоненти в єдиний інформаційний простір.

Так, активне застосування цифрових двійників, тобто віртуальних моделей фізичних об'єктів або процесів, які відтворюють їхній стан у реальному часі стало характерною складовою сучасного напрямку автоматизації. Зокрема, цифрові двійники дозволяють моделювати, прогнозувати відмови, оптимізувати параметри, а також розробляти сценарії без зупинки реального виробництва. Власне застосування цієї технології суттєво знижує витрати на експлуатацію, а також підвищує ефективність керування.

З огляду на сказане вище, сучасні підходи до автоматизації виробничих процесів реалізують інтеграцію SCADA, MES, ERP-рішень із IoT-платформами та цифровими двійниками в єдине кіберфізичне середовище. Така інтеграція стає передумовою для впровадження інтелектуального виробництва, на якому всі рівні – від сенсорів до корпоративних систем – функціонують узгоджено, гарантуючи повну гнучкість і прозорість керування.

### 1.3 Проблеми інформаційно-технічного забезпечення кіберфізичних систем

З огляду на те, що технології автоматизації та цифровізації набули стрімкого розвитку, проте впровадження КФС у виробництво водночас породжує низку проблем інформаційно-технічного характеру, які спричиняють істотний вплив на їхню ефективність та надійність, а також безпеку. Такі

проблеми можна обумовити технічними, і організаційними чинниками, що пов'язані з інтеграцією різнорідних елементів до єдиної інтелектуальної системи.

### 1.3.1 Низький рівень інтеграції між різними підсистемами

Багато підприємств устатковують різнотипне обладнання та програмні засоби, що функціонують, базуючись на закритих або несумісних протоколах обміну даними. Це породжує суттєві труднощі для створення єдиного інформаційного середовища, в якому всі компоненти (від сенсорів до корпоративних систем) взаємодіяти можуть безпосередньо. До того ж, відсутність стандартизованих інтерфейсів і протоколів (на кшталт OPC UA, MQTT) зумовлює фрагментацію даних, а, отже, призводить до втрати їхньої цілісності у разі передавання між рівнями системи [11].

### 1.3.2 Проблеми оброблення великих обсягів даних

Зазвичай КФС генерує значні потоки інформації з тисяч сенсорів і контролерів у реальному часі. Загальноприйняті системи зберігання й оброблення не завжди спроможні підтримувати швидкодію та масштабованість, які необхідні. Втім головна проблема полягає у забезпеченні оперативної аналітики, тобто рішення повинні прийматися за частки секунди. У результаті виникає потреба у залученні технологій edge computing (обчислення на периферії) та Cloud Computing з метою балансування навантаження між локальними і хмарними ресурсами [11].

### 1.3.3 Забезпечення кібербезпеки

На сьогодні ризик кібератак або несанкціонованого доступу разом із втратою критично важливої інформації через зростання кількості підключених пристроїв значно підвищується.

У зв'язку з цим до основних загроз належать:

– перехоплення чи модифікація даних під час передавання;

- зараження шкідливим програмним забезпеченням (ПЗ) промислових контролерів;
- несанкціоноване втручання до алгоритмів керування.

З метою подолання перелічених ризиків доречно встановити багаторівневу систему захисту, котра, зі свого боку, містить автентифікацію, контроль доступу, шифрування, а також моніторинг подій безпеки.

#### 1.3.4 Проблеми синхронізації фізичних і цифрових процесів

Головну ідею КФС визначають як взаємодію реального виробництва з його цифровим відображенням (цифровим двійником). Утім можуть виникати труднощі з узгодженням часових затримок або різницею між фізичними вимірюваннями та модельними даними. За таких обставин це може призводити до некоректного керування. Отже, доцільно забезпечити синхронізацію потоків даних у реальному часі, а це, зі свого боку, потребує не лише високоточного часу та стабільних мереж, але й ефективних протоколів передавання.

#### 1.3.5 Високі вимоги до надійності та відмовостійкості

З огляду на те, що КФС функціонують у режимі безперервного керування технологічними процесами, то навіть нетривалий збій може спричинити аварії чи значні економічні втрати.

У зв'язку з цим постає потреба у розробленні резервованих систем. Завдяки їм можна автоматично відновлювати працездатність після відмов. Окрім того, проводити предиктивне обслуговування, аналізуючи стан обладнання.

#### 1.3.6 Кадрові й організаційні проблеми

Упровадження КФС, безумовно, потребує фахівців із міждисциплінарними знаннями, зокрема, у галузі ІТ, аналізу даних, кібербезпеки й автоматизації, адже недостатня кваліфікація працівників або

навіть брак чіткої стратегії цифрової трансформації підприємства спроможні знизити ефективність функціонування систем, які розробляються.

З огляду на зазначене вище, нагальними проблемами інформаційно-технічного забезпечення КФС будуть забезпечення надійного обміну даними, захист інформації, інтеграція різнорідних компонентів, синхронізація процесів, а також кадрове забезпечення. Таким чином, подолати окреслені проблеми цілком можливо шляхом розроблення комплексних систем автоматизації, котрі, зі свого боку, базуються на принципах безпечної архітектури, стандартизованих протоколах і розподіленому обробленні даних.

#### 1.4 Формулювання вимог до автоматизованої системи

За результатами проведеного аналізу предметної області, структури кіберфізичних систем і проблем їхнього інформаційно-технічного забезпечення було визначено комплекс вимог до автоматизованої системи. Вона повинна забезпечувати ефективну інтеграцію, контроль і керування процесами у межах кіберфізичного виробництва.

Зі свого боку, такі вимоги поділяються на функціональні, не функціональні й експлуатаційні [12].

##### 1.4.1 Функціональні вимоги

Функціональні вимоги окреслюють ключові можливості автоматизованої системи, котрі забезпечують виконання завдань, покладених на неї [13]:

- збір і оброблення даних у реальному часі. Так, за допомогою системи має виконуватись безперервний збір інформації від сенсорів, контролерів і пристроїв різних типів, водночас повинна проводитись попередня фільтрація, нормалізація та збереження даних у централізованому сховищі;

- моніторинг і візуалізація процесів. У такий спосіб забезпечується графічне відображення поточного стану технологічних процесів, параметрів

роботи обладнання, а також формуються попередження про відхилення від нормальних режимів;

- інтелектуальне керування. Насамперед система повинна послуговуватися алгоритмами адаптивного керування, прогнозувати відмови, оптимізувати параметри технологічних процесів відповідно до аналізу історичних і поточних даних;

- підтримка прийняття рішень. Мова йде про передбачення модулів аналітики, що дозволяють оцінювати ефективність виробництва, прогнозувати та формувати рекомендації для оператора, до того ж автоматично ініціювати зміни в роботі системи;

- інтеграція з чинними виробничими системами. Зі свого боку, система повинна підтримувати взаємодію з SCADA-, MES- і ERP-рівнями, зовнішніми БД, а також IoT-платформами через стандартизовані протоколи (OPC UA, MQTT, REST API тощо);

- архівація та звітність. Мається на увазі забезпечення можливості збереження історичних даних, окрім того, формування звітів і надання користувачу аналітичних відомостей про ефективність роботи системи.

#### 2.4.2 Нефункціональні вимоги

Нефункціональні вимоги, як правило, окреслюють якісні характеристики системи, котрі впливають на її зручність використання, продуктивність та надійність [13]:

- надійність та відмовостійкість. Системі необхідно забезпечувати стабільну роботу, незважаючи на часткові збої окремих компонентів, а, отже, мати механізми автоматичного відновлення працездатності;

- швидкодія. Час реакції параметри процесу, котрі зазнали змін, не повинен бути довшим за допустимі межі для конкретної технології;

- масштабованість. Архітектура системи повинна бути підготовленою для, збільшення кількості підключених пристроїв і користувачів, а, отже, і розширення функціональності без істотних змін структури;

- інформаційна безпека. Спроможність забезпечувати, контроль доступу до ресурсів та автентифікацію користувачів, шифрування переданих даних і наявність журналу подій безпеки;
- сумісність і відкритість. Системі необхідно підтримувати стандарти промислових комунікацій, а також бути відкритою для інтеграції з програмними продуктами інших розробників;
- зручність користування. Інтерфейс користувача повинен бути інтуїтивно зрозумілим, а також надавати можливість налаштувати панелі моніторингу та доступу до функцій, зважаючи на ролі оператора.

#### 1.4.3 Експлуатаційні вимоги

Експлуатаційні вимоги мають такі пункти [13]:

- мінімізувати потреби щодо ручного втручання. Система повинна забезпечувати максимально можливий рівень автоматизації операцій, оже, вплив людського чинника – мінімальний;
- підтримувати безперервне функціонування. Бути спроможною працювати в цілодобовому режимі, а також оновлювати програмні компоненти без зупинки системи;
- документувати та обслуговувати. Система повинна містити засоби ведення технічної документації, сервісного обслуговування та журналів подій.

Сформульовані вище вимоги можна розглядати як підґрунтя для подальшого вдосконалення архітектури, а також програмно-апаратної реалізації автоматизованої системи, котра, зі свого боку, зорієнтована на забезпечення ефективної, стабільної, безпечної роботи кіберфізичного виробництва.

#### 1.5 Висновки до першого розділу

За результатами проведеного аналізу предметної області кіберфізичних виробничих систем і вивчення сучасних інформаційно-технічних рішень можемо підсумувати, що КФС набули значення ключової технологічної

платформи Індустрії 4.0, зокрема, вона запроваджує інтеграцію реальних фізичних процесів із цифровими моделями й аналітичними сервісами, а також засобами автоматизації. Так, вони можуть суміщати сенсорні пристрої разом із мережевою інфраструктурою, обчислювальними модулями й інтелектуальними алгоритмами та системи керування в єдиному комплексі.

Під час розгляду було встановлено, що провідні промислові платформи, на кшталт Siemens MindSphere, AVEVA System Platform, Ignition SCADA, Bosch IoT Suite, надають високий рівень функціональності, проте їм властиві й певні обмеження, що окреслюються вартістю, труднощами під час інтеграції, залежністю від постачальника разом із обмеженою гнучкістю у разі адаптації під конкретні виробничі процеси.

Опрацювання сучасних технологій обміну даними підтвердило, що протоколи MQTT, OPC UA та REST API є актуальними для кіберфізичних виробництв, адже реалізують надійність, низьку затримку, масштабованість, а також можливість інтеграції з різнорідними пристроями.

Окрім того, було доведено, що застосування хмарних технологій разом із контейнеризацією посідають роль ключових чинників щодо підвищення гнучкості та масштабованості інформаційно-технічних систем, водночас, спрощує їхнє налаштування, супровід і розгортання.

За результатами аналізу було встановлено, що чинні рішення, як правило, охоплюють збір даних і візуалізацію, проте інтеграція реального часу з інтелектуальною аналітикою й адаптивним керуванням досі недостатньо реалізовані. З огляду на це, постає потреба щодо розроблення систем нового покоління.

Було доведено необхідність побудови власної автоматизованої системи інформаційно-технічного забезпечення. У ній повинні поєднуватись висока швидкодія та аналітичні можливості, підтримка сучасних протоколів і здатність до адаптивного керування в умовах кіберфізичного виробництва.

На тлі проведеного аналізу було визначено пріоритетні вимоги до майбутньої системи, зокрема, мова йде про модульність, підтримку реального

часу, масштабованість, відмовостійкість, інтеграцію алгоритмів прогнозування та виявлення аномалій, послуговування відкритими технологіями, а також мінімальну залежність від ліцензійних обмежень.

Отже, відповідно до розділу 1 упорядковано цілісне уявлення про сучасний стан кіберфізичних систем, охарактеризовано їхні сильні та слабкі сторони й аргументовано доцільність розроблення власної автоматизованої системи. Результати, що були отримані, визначено як підґрунтя для створення концепції й архітектури розробленої системи, котру реалізовано у наступних розділах роботи.

## 2 РОЗРОБЛЕННЯ КОНЦЕПЦІЇ ТА АРХІТЕКТУРИ АВТОМАІЗИЗОВАНОЇ СИСТЕМИ

Мету автоматизації кіберфізичного виробництва можна окреслити як створення інтегрованої системи керування, що проводить аналіз та оптимізацію, контроль та оперативний моніторинг усіх етапів виробничого процесу шляхом об'єднання фізичних і цифрових компонентів до єдиного інформаційного середовища.

Власне завдання автоматизації доречно сформулювати як забезпечення взаємодії між обчислювальними модулями, контролерами, сенсорними пристроями, БД із інтерфейсами користувача. Тобто керувати процесами в реальному часі можна з мінімальним втручанням оператора [14].

### 2.1.1 Основна мета автоматизації

Ключова мета автоматизації полягає у розробленні системи, функції якої відповідають за:

- збір, оброблення та зберігання інформації щодо стану виробничих об'єктів;
- адаптивне керування технологічними процесами з огляду на поточні умови;
- прогнозування та попередження щодо відмов обладнання;
- підтримка інформаційного обміну між усіма рівнями кіберфізичного виробництва;
- забезпечення високого рівня безпеки, а також надійності функціонування.

### 2.1.2 Основні завдання автоматизованої системи

Автоматизований моніторинг стану виробничого середовища має на меті збирання даних від сенсорів, які контролюють фізичні параметри як-от: температуру, тиск, вібрації, споживання енергії тощо.

Передавання даних у реальному часі реалізується через промислові мережі (Ethernet/IP, ModbusTCP, Profinet, MQTT). Таким чином, можна передавати інформацію між фізичним рівнем і цифровими компонентами [15].

Аналітичне оброблення та прийняття рішень здійснюється за допомогою алгоритмів аналізу даних, а також машинного навчання з метою виявлення закономірностей, прогнозування несправностей і, насамперед, підвищення ефективності виробництва.

Інтелектуальне керування процесами спирається на аналіз інформації, після чого система формує керувальні дії для регулювання технологічних параметрів, забезпечуючи, таким чином, стабільність процесів, а також економію ресурсів.

Інформаційно-технічна інтеграція полягає в тому, що система необхідно бути сумісною з популярними виробничими платформами (MES, ERP, SCADA), отже, підтримувати відкриті стандарти обміну даними.

Забезпечення кібербезпеки реалізується шляхом упровадження захищених каналів зв'язку, аутентифікації користувачів, контролю доступу та шифрування даних.

### 2.1.3 Очікувані результати автоматизації

За результатами впровадження системи очікується:

- зменшення часу реагування на зміни технологічних параметрів;
- зниження впливу людського чинника під час керування виробничими процесами;
- підвищення ефективності використання ресурсів (енергії, матеріалів, часу);

- оптимізація технічного обслуговування шляхом проведення предиктивного аналізу стану обладнання;
- створення бази для подальшого впровадження технологій Індустрії 4.0, на кшталт цифрових двійників і хмарного керування виробництвом.

Підсумуємо, що постановка задачі автоматизації має на меті створення гнучкої, масштабованої та безпечної КФС, котра спроможна реалізувати інтеграцію фізичних процесів і цифрових технологій на базі єдиного інтелектуального виробничого середовища.

## 2.2 Вибір принципів побудови автоматизованої системи

Проектування автоматизованої системи кіберфізичного виробництва доцільно створювати на сукупності принципів, які гарантують її ефективність, гнучкість, масштабованість, а також безпеку. Такі принципи окреслюють загальні підходи щодо взаємодії компонентів, побудови архітектури й організації інформаційних потоків у межах системи.

Система автоматизації повинна характеризуватися модульною структурою, елементами якої є розподіл функцій між окремими компонентами (сенсорними модулями, аналітичними сервісами та контролерами). Модульність реалізує власне розширення функціоналу та простоту модернізації, можливість заміни або оновлення окремих елементів без зупинки всієї системи, гнучкість адаптації до змін технологічного процесу [13-16].

Ієрархічність надає можливість розподілити систему на рівні керування, зокрема, від польового (сенсори, приводи) до корпоративного рівня (аналітика, планування, ERP). Зазначений підхід, зі свого боку, спрощує організацію потоків даних і сигналів керування.

Утім ефективне функціонування КФС реалізується лише за умови повної інтеграції фізичних і цифрових компонентів.

Пояснимо більш детально: всі елементи, тобто аналітичні модулі, сенсори, контролери та БД, повинні функціонувати у спільному

інформаційному просторі з узгодженими протоколами обміну. Для цього, як правило, послуговуються відкритими стандартами зокрема, OPCUA, MQTT, RESTAPI, JSON, XML, які відповідають за сумісність і масштабованість системи незалежно від виробника обладнання.

Так, для сучасних КФС пріоритетним вважається підхід децентралізації, коли прийняття рішень виконується власне на локальних рівнях, тобто у контролерах або мікропроцесорних вузлах. Таким чином, мова йде про зменшення затримки під час обміну даними, підвищення відмовостійкості системи, а також забезпечення безперервного функціонування навіть у разі порушення зв'язку з центральним сервером.

Децентралізоване керування, насамперед, поєднується з глобальним моніторингом, що створює розподілену інтелектуальну систему керування [16].

Систему доцільно розробляти за принципом відкритої архітектури завдяки чому забезпечується можливість додавання нових пристроїв, модулів чи сервісів без значних змін структури. Водночас відкритість запроваджує довготривалу життєздатність системи та гарантує сумісність із майбутніми технологіями, а масштабованість – потенціал для розвитку, зважаючи на потреби підприємства.

З огляду на те, що систему автоматизації задіяно у середовищі з інтенсивним обміном даними, то значну увагу доцільно приділити захисту інформації, зокрема:

- автентифікації користувачів і пристроїв;
- захисту від несанкціонованого доступу;
- шифруванню даних під час передавання;
- моніторингу подій безпеки та ведення журналів активності.

Дотримання складових цього принципу є обов'язковим для запобігання кібератакам, а також збереження цілісності виробничих даних.

Так, система має бути спроможною адаптуватися до змін зовнішніх умов і технологічних параметрів. Задля цього доречним буде впровадження алгоритмів самонавчання, аналізу даних у реальному часі та предиктивного

керування, адже вони дозволяють оптимізувати роботу системи без прямого втручання оператора. До того ж, послуговування технологіями машинного навчання та штучного інтелекту забезпечить перехід від реактивного керування до проактивного.

Окрім того, система повинна стабільно працювати навіть під час часткових збоїв. З цією метою задіюють контроль працездатності модулів у реальному часі, резервування каналів зв'язку та серверів, й автоматичне відновлення після відмов, оскільки це не лише гарантує безперервність технологічного процесу, але й запобігає аварійним ситуаціям.

Підкреслимо, що наявність засобів діагностики, журналів подій і можливостей дистанційного обслуговування передбачає створення інтуїтивно зрозумілого інтерфейсу користувача. Даний підхід власне забезпечує оперативне реагування на відхилення в роботі системи разом із ефективною роботою персоналу.

Таким чином, окреслені принципи побудови автоматизованої системи визначають її архітектурну модель, гнучкість і стійкість до зовнішніх впливів, рівень інтеграції, адже є концептуальним підґрунтям для подальшого розроблення структури та функціональних модулів системи кіберфізичного виробництва.

### 2.3 Розроблення загальної архітектури

Архітектура автоматизованої системи кіберфізичного виробництва передусім окреслює структурну організацію апаратних і програмних компонентів, їхні взаємозв'язки, канали обміну даними, а також логіку взаємодії між рівнями керування.

Мету розроблення архітектури визначено як створення гнучкої, масштабованої та відмовостійкої системи, котра спроможна реалізувати синхронізацію фізичних процесів із цифровими моделями у реальному часі.

Архітектуру системи розроблено на базі багаторівневої моделі керування, що, зі свого боку, передбачає ієрархічну організацію елементів відповідно до їхніх функцій [16].

Виділяють 5 рівнів системи:

- польовий рівень (Sensor/ActuatorLayer) містить приводи, сенсори, виконавчі механізми, котрі реалізують безпосередню взаємодію з фізичними об'єктами виробництва;

- рівень керування (ControlLayer), елементами якого є мікроконтролери чи програмовані логічні контролери (PLC), або промислові комп'ютери, котрі розраховані на задіяння алгоритмів регулювання та локального оброблення даних;

- рівень моніторингу та візуалізації (SupervisoryLayer), містить системи SCADA, НМІ й інтерфейси операторів, які відповідають за контроль процесів, а також збирання статистики;

- рівень керування виробництвом (ManufacturingExecutionLayer)– системи MES, зі свого боку, виконують планування, диспетчеризацію й аналіз виробничих показників;

- корпоративний рівень (EnterpriseLayer) характеризується системи ERP, CRM, PLM, призначення яких – інтегрування виробничих даних у бізнес-процеси підприємства.

Окрім того, до загальної архітектури автоматизованої системи належать такі логічні компоненти:

- підсистема збору даних насамперед приймає сигнали від сенсорів, потім виконує їхнє попереднє оброблення із подальшим передаванням до контролерів або серверів;

- підсистема керування містить алгоритми регулювання технологічних параметрів, підтримує задані режими роботи, а також забезпечує стабільність процесів;

- підсистема аналітики відповідає за оброблення великих масивів даних (BigData), побудову прогнозів разом із оптимізацією параметрів роботи обладнання;

- комунікаційна підсистема реалізує обмін даними між усіма рівнями шляхом залучення промислових протоколів (Modbus, MQTT, OPCUA, Ethernet/IP);

- підсистема інформаційної безпеки характеризується моніторингом подій безпеки, функціями автентифікації, шифрування, контролю доступу;

- підсистема візуалізації власне надає користувачеві необхідні інструменти задля спостереження за технологічними процесами і, водночас, керування ними.

Як правило, до елементів апаратної частини належать:

- вимірювальні пристрої, зокрема, сенсори температури, тиску, струму, вібрацій тощо;

локальні обчислювальні вузли, які, зі свого боку, відповідають за попередню аналітику;

- центральний сервер або хмарне середовище, котрі необхідні для зберігання, оброблення й аналітики даних;

- контролери задля оброблення сигналів, а також керування виконавчими механізмами;

- мережева інфраструктура, котра має на меті підтримання промислових протоколів, а також забезпечення стабільного зв'язку між компонентами.

До того ж, вагомими компонентами є резервування живлення, подвійні канали зв'язку та засоби безперервного моніторингу стану обладнання.

ПЗ системи власне реалізує:

- збір і зберігання даних у реальному часі (за допомогою MQTT-брокерів або OPC-серверів);

- аналітичне оброблення (зокрема шляхом застосування алгоритмів машинного навчання з метою прогнозування станів);

- візуалізацію процесів (завдяки веб- або десктоп-інтерфейсів);

- синхронізацію фізичних і цифрових моделей виробництва (технологія «цифрового двійника»);

- підтримку API для інтеграції з зовнішніми системами керування.

З метою підвищення ефективності доцільно задіяти розподілену архітектуру типу client-server або cloud-edge hybrid, на базі якої частина обчислень виконується локально, а інша – у хмарному середовищі.

Передавання даних, зі свого боку, у системі функціонує двонаправлено: знизу вгору – від сенсорів до серверів для моніторингу, оброблення й аналізу, а також з гори вниз – від систем керування до виконавчих пристроїв для регулювання процесів.

Комунікаційна інфраструктура за своїм призначенням полягає у забезпеченні мінімальних затримок передавання, високій пропускну здатності каналів, захисті від втрати пакетів, а також містить стандартизовані протоколи взаємодії між усіма елементами.

На противагу типовим автоматизованим системам, архітектура КФС передбачає щільну інтеграцію фізичного рівня з обчислювальними моделями, чинного зворотного зв'язку в реальному часі між об'єктом керування та цифровим середовищем, використання інтелектуальних агентів, які спроможні до самостійного приймання рішень, а також характеризується підтримкою самодіагностики разом із самовідновленням системи.

Підсумуємо, що створена архітектура поєднує властивості традиційних автоматизованих систем керування із інтелектуальними технологіями Індустрії 4.0, утворюючи, зі свого боку, підґрунтя для побудови сучасного кіберфізичного виробництва.

#### 2.4 Забезпечення інтеграції з зовнішніми системами

Вагомим аспектом щодо ефективного функціонування кіберфізичної автоматизованої системи визначено забезпечення інтеграції з зовнішніми інформаційними системами та системами керування. Така інтеграція породжує

єдиний інформаційний простір підприємства, в межах якого злагоджено функціонують адміністративні, аналітичні, технологічні, та корпоративні рівні керування.

Мету інтеграції доречно визначити як забезпечення безперервного обміну даними між системами різного рівня, йдеться про обладнання від польового рівня до хмарних сервісів і корпоративних платформ (на кшталт ERP, MES, SCADA, PLM тощо) [17].

Приклад інтеграції з зовнішніми системами відображено на рисунку 2.1.

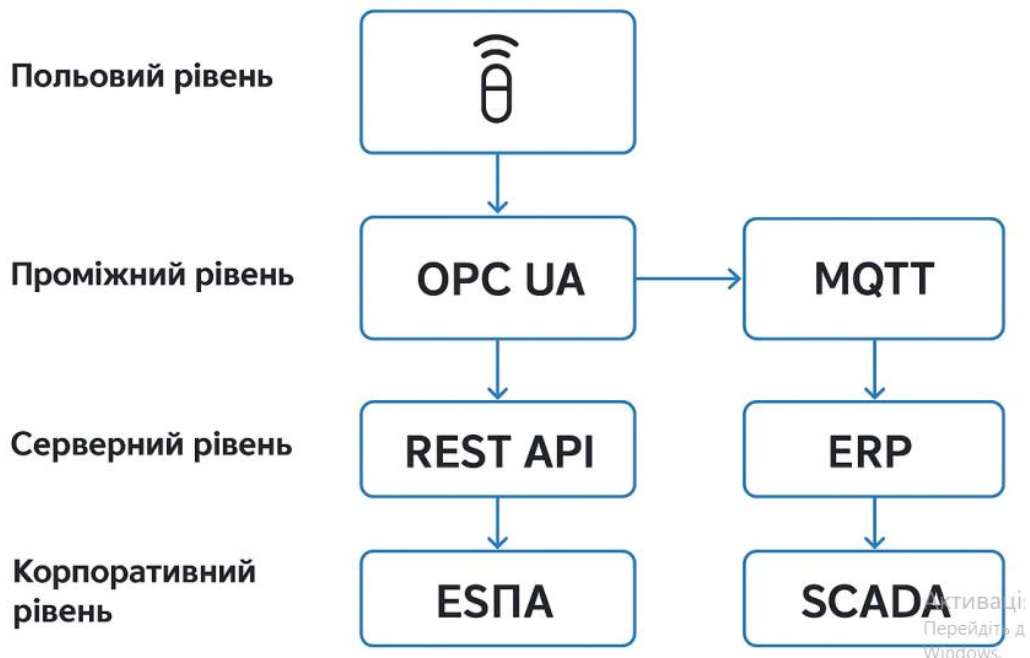


Рисунок 2.1 – Інтеграція з зовнішніми системами

Як правило, інтеграція діє на базі таких принципів:

- відкритості, тобто за основу беруться загальновизнані стандарти та протоколи обміну;
- масштабованості, тобто спроможності розширювати систему без зміни її архітектури;
- сумісності, що забезпечує можливість взаємодії систем різних виробників;

- реального часу, що полягає у мінімальній затримці під час обміну інформацією;
- безпечності, мається на меті захист даних, які були передані, від несанкціонованого доступу;
- реальний час – мінімальна затримка при обміні інформацією.

З метою реалізації інтеграції діють універсальні промислові протоколи, які, насамперед, підтримують сучасні архітектури КФС.

Вважається, що OPC UA є головним стандартом інтеграції промислових систем. Він, зі свого боку, забезпечує уніфіковану модель даних для контролерів, сервісів пристроїв, а також реалізує двосторонній обмін інформацією між польовим і корпоративним рівнями, безпечно передавання, використовуючи вбудовані механізми шифрування, автентифікації та контролю доступу та незалежність від платформи та операційної системи. OPC UA є основою для створення єдиного інформаційного простору підприємства, де всім компонентам властива спільна структура даних.

MQTT називають легковаговий мережевий протокол, який оптимізовано для роботи в системах реального часу з великою кількістю пристроїв. Перелічимо особливості мережевого протоколу:

- мінімальні вимоги до пропускної здатності мережі;
- підтримка шифрування та перевірки цілісності повідомлень;
- модель комунікації «публікація-підписка»;
- можливість підключення IoT-пристроїв, сенсорів, контролерів.

MQTT активно залучається для суміщення кіберфізичних пристроїв із хмарними аналітичними платформами.

REST API власне реалізує інтеграцію на рівні програмного обміну між системами. Окреслимо його переваги:

- можливість інтеграції з вебдодатками, ERP- та MES-системами;
- обмін даними у форматах JSON або XML;
- використання стандартних веб-протоколів (HTTP, HTTPS);
- простота реалізації та масштабування.

REST API визначено як головний інтерфейс для взаємодії між програмними компонентами автоматизованої системи із зовнішніми сервісами.

Так, інтеграція систем підпорядковується багаторівневій архітектурі:

- польовий рівень – сенсори, контролери, приводи;
- проміжний рівень – збір, попереднє оброблення, а також агрегація даних;
- серверний рівень – передавання даних до системи керування шляхом послуговування OPC UA/MQTT;
- корпоративний рівень – інтеграція з ERP, SCADA, MES, аналітичними платформами.

Розглянута модель сприяє забезпеченню логічної взаємодії між фізичними процесами й інформаційними системами підприємства.

У цілому функціонування інтеграційної підсистеми, що ґрунтується на окреслених технологіях забезпечує:

- скорочення часу обміну інформацією та прийняття рішень;
- уніфікацію потоків даних між усіма рівнями керування;
- можливість централізованого моніторингу й аналітики;
- підвищення прозорості виробничих процесів;
- підготовку до переходу на стандарти Індустрії 4.0.

За традиційним сценарієм система автоматизації виробничої лінії надає дані до сервера аналітики про стан обладнання, використовуючи протокол OPC UA, який через REST API взаємодіє з корпоративною ERP-системою. Водночас контролери надсилають телеметрію до хмарного сервісу MQTT. Такі дії, зі свого боку, дозволяють відстежувати продуктивність обладнання у реальному часі з будь-якої точки доступу.

З огляду на все сказане вище, можемо резюмувати, що інтеграція з зовнішніми системами гарантує високу взаємодію, адаптивність та керованість кіберфізичного виробництва, створюючи, насамперед, базу для побудови розумного підприємства (SmartFactory).

## 2.5 Інформаційна модель

Здебільшого інформаційну модель автоматизованої системи інформаційно-технічного забезпечення кіберфізичного виробництва розглядають як формалізований опис процесів зі збору й оброблення, передавання та використання даних, які функціонують у межах системи. Таким чином, мета її створення окреслюється забезпеченням цілісності разом із узгодженістю й ефективним обміном інформацією між усіма компонентами кіберфізичного середовища [17].

Так, інформаційна модель, насамперед, відображає взаємозв'язок як між фізичними об'єктами, так і програмними компонентами з користувачами, котрі є складовими виробничого процесу. До того ж, вона окреслює структуру інформаційних потоків, а також рівні доступу, формати даних і правила обміну.

### 2.5.1 Структура інформаційної моделі

Інформаційну модель автоматизованої системи, приклад якої наведено на рисунку 2.2, побудовано на базі багаторівневого принципу, що, зі свого боку, відображає логічну взаємодію між ключовими компонентами системи, зокрема [17]:

- сенсорний рівень задіяно для збору даних із фізичних об'єктів виробництва;
- комунікаційний рівень полягає у передаванні даних між контролерами, пристроями сенсорними, серверами та хмарними платформами;
- аналітичний рівень спрямовано на аналіз, а також оброблення та зберігання інформації;
- рівень керування та інтерфейсу користувача відповідає за взаємодію оператора із системою з одного боку, та за прийняття рішень і відображення результатів – з іншого.



Рисунок 2.2 – Інформаційна модель автоматизованої системи

### 2.5.2 Опис рівнів інформаційної моделі

На сенсорному рівні робота ґрунтується на безперервному збиранні даних із фізичних процесів завдяки сенсорам і вимірювальним пристроям. Зазвичай, надходить інформація про енергоспоживання, температуру, тиск, оберти двигунів, рівень вібрацій тощо.

Передавання даних виконується шляхом задіяння стандартних промислових протоколів, як-от: Modbus, CAN, OPCUA. Втім задля підвищення достовірності передбачено залучення й алгоритмів фільтрації разом із калібруванням сигналів.

Комунікаційний рівень діє, передусім, з метою забезпечення безпечного та надійного обміну даними між усіма компонентами системи. На цьому рівні послуговуються дротовими (Ethernet, RS-485), а також бездротовими каналами зв'язку (Wi-Fi).

З метою гарантування цілісності даних залучається маршрутизація з контролем доступу та шифруванням трафіку (SSL/TLS). До того ж, на комунікаційному рівні активується механізм «публікація-підписка», за

допомогою якого масштабувати систему можна без втрати продуктивності.

Аналітичний рівень, перш за все, є ядром системи, в межах якого виконуються такі дії:

- попереднє оброблення даних, зокрема, очищення, нормалізація й агрегація;
- зберігання інформації у реляційних або хмарних БД (SQL/NoSQL);
- формування рекомендацій щодо оптимізації виробництва;
- аналіз і прогнозування шляхом алгоритмів машинного навчання.

Окрім того, на аналітичний рівень покладено функцію цифрового двійника виробничої системи, завдяки цьому можна моделювати сценарії роботи без втручання до фізичного середовища.

Рівень керування та інтерфейсу користувача насамперед забезпечує інтерактивну взаємодію оператора з системою автоматизації.

До головних функцій даного рівня належить:

- віддалене керування процесами;
- відображення ключових показників ефективності (KPI);
- налаштування параметрів і сценаріїв роботи;
- моніторинг у реальному часі стану виробництва.

Так, інтерфейс функціонує як вебзастосунок або НМІ-панель, таким чином, зручний доступ до всіх функцій системи надається через графічні елементи керування.

Втім інформаційну взаємодію між рівнями реалізовано відповідно до принципу DataFlow (знизу вгору), де сенсорні дані передаються до аналітичного рівня задля оброблення й прийняття рішень, на відміну від принципу ControlFlow (згори вниз), де керувальні команди надходять від рівня керування до виконавчих пристроїв [17].

Отже, створюється замкнений інформаційно-керувальний контур, який реалізує безперервну адаптацію системи до змін виробничого середовища.

Підкреслимо, що модель, яку було запропоновано, забезпечує:

- єдиний інформаційний простір для всіх учасників виробництва;

- можливість щодо масштабування й інтеграції нових пристроїв;
- підвищення надійності шляхом розподіленого оброблення даних;
- скорочення часу реагування на події;
- підтримку рішень у реальному часі із засосуванням гібридної архітектури (локальних і хмарних ресурсів).

## 2.6 Висновки до другого розділу

У другому розділі наведено комплексне аргументування концепції, структурних принципів, а також архітектурних рішень автоматизованої системи інформаційно-технічного забезпечення кіберфізичного виробництва.

Розроблено концептуальну модель автоматизованої системи, в котрій поєднуються аналітичні алгоритми, комунікаційна інфраструктура, сенсорні модулі, модуль керування та підсистема взаємодії з виробничим обладнанням. Власне модель утворено на парадигмі КФС відповідно до принципів Індустрії 4.0.

Упорядковано архітектуру системи, зважаючи на модульний та сервісно-орієнтований підходи (SOA), тобто забезпечується гнучкість і розширюваність, можливість масштабування й оновлення функціональних модулів без зупинки роботи всього комплексу.

Окреслено функціональні вимоги до системи. Власне акцент робиться на зборі та передаванні, обробленні й аналізі, візуалізації технологічних даних у режимі реального часу. До того ж, нагальним стає питання щодо забезпечення автоматизованого прийняття рішень та формування впливів керування.

Доведено вибір протоколів обміну даними (MQTT, OPC UA, REST API), актуальність яких полягає у забезпеченні низької затримки, надійної синхронізації, безпечному передаванні даних, а також сумісності із значним різноманіттям промислових пристроїв і контролерів.

Розроблено логічну структуру даних та інформаційні потоки. Вони, зі

свого боку, визначають взаємодію між сенсорами, брокерами, аналітичними сервісами, контролерами та виконавчими механізмами. Таким чином, можна формалізувати зв'язки між підсистемами та, водночас, реалізувати однозначність логіки роботи.

Використання контейнеризації, що відповідає за ізолюваність сервісів, швидке масштабування, портативність платформи та зручність розгортання системи на будь-яких виробничих потужностях теж обґрунтовано.

Окрім того, побудовано діаграми взаємодії, потоків даних, станів, а також архітектурні схеми, котрі демонструють логіку роботи системи, взаємозв'язок модулів і ролеву структуру інформаційних потоків у кіберфізичному середовищі.

За результатами сформованої концепції й архітектури доведено, що розроблена система втілює багаторівневу взаємодію між фізичними та цифровими компонентами, підтримує аналітику та прогностичне керування, а, отже, стає гнучкою та модульною, а також є вигідною до інтеграції в сучасні виробничі процеси.

Другий розділ охопив фундаментальні принципи й архітектурні рішення разом із концептуальними засадами автоматизованої системи. Таким чином, було забезпечено підґрунтя задля подальшого математичного моделювання та реалізації алгоритмів, а також тестування системи, що будуть розглянуті у наступних розділах.

### 3 МОДЕЛЮВАННЯ ФУНКЦІОНУВАННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ

Моделювання функціонування автоматизованої системи інформаційно-технічного забезпечення кіберфізичного виробництва – вагомий етап дослідження. Завдяки цьому можна комплексно оцінити як динаміку технологічних процесів, логіку роботи програмних модулів, так і відмовостійкість, точність алгоритмів разом із швидкодією підсистем. Залучення математичних моделей стає засобом для створення формалізованого підґрунтя для дослідження, оптимізації, тестування, а також порівняння розробленої системи з промисловими рішеннями, котрі наразі є актуальними [18-21].

Таким чином, даний розділ зорієнтовано на виконання комплексу змодельованих досліджень.

#### 3.1 Моделювання роботи системи

Моделювання роботи системи реалізовано відповідно до такої моделі: “DAQ → Communication → Analytics → Control → Field → Feedback”.

За результатами проведеного моделювання вдалося оцінити:

- точність регулювання;
- стійкість до шумів і впливів;
- час реакції системи у режимі реального часу;
- ефективність прогнозних моделей;
- поведінку системи через втрату пакетів або відмову обладнання.

Окрім того було опрацьовано шість пріоритетних сценаріїв функціонування системи: штатний режим, випадкові впливи, шумові сигнали, різкі стрибки параметрів, аномалії, аварійні відмови.

Власне результати, котрі було отримано під час моделювання роботи системи, зведено у таблицю 3.1.

Таблиця 3.1 – Отримані результати роботи системи

Параметр	Значення
час повного циклу керування	120 – 180 мс
точність ML-прогнозування	94 – 96 %
відновлення після впливів	1,2 – 1,8 с
час реакції аварійного режиму	< 120 мс

Таким чином, моделювання підтвердило попередні висновки щодо стабільності системи, а також відповідність вимогам кіберфізичного виробництва.

## 3.2 Побудова математичних моделей

### 3.2.1 Математична модель технологічного процесу

Для опису технологічного параметру оберемо, наприклад, динамічну змінну  $x(t)$ . У такому разі динаміку процесу доцільно відтворити за допомогою диференціального рівняння [18-21]:

$$\frac{dx(t)}{dt} = f(x(t), u(t), d(t)), \quad (3.1)$$

де  $u(t)$  – вплив керування;

$d(t)$  – вплив (шум, нестабільність навантаження).

Лінійне наближення об'єкта тоді подамо так:

$$dx(t)dt = -ax(t) + bu(t) + d(t), \quad (3.2)$$

де  $a, b > 0$  – коефіцієнти динаміки об'єкта.

Передаточна функція об'єкта  $W(s)$  зі свого боку, обчислюватиметься як:

$$W(s) = \frac{b}{s+a}. \quad (3.3)$$

### 3.2.2 Модель сенсорної підсистеми

Припустімо, що вимірний сигнал  $y(t)$  можемо оформити так:

$$y(t) = x(t) + n(t), \quad (3.4)$$

де  $n(t)$  – шум сенсора.

Утім для зменшення шумової складової доцільно залучити лінійні та нелінійні фільтри [18-21].

Фільтр ковзного середнього:

$$\hat{x}(k) = \frac{1}{N} \sum_{i=0}^{N-1} y(k-i). \quad (3.5)$$

Експоненційний фільтр:

$$\hat{x}(k) = \alpha y(k) + (1 - \alpha) \hat{x}(k-1). \quad (3.6)$$

### 3.2.3 Математична модель комунікаційного каналу

Комунікаційний канал із затримкою  $\tau$  на тлі ймовірної втрати пакета  $p$  опишемо як [18-21]:

$$y_{tr}(t) = y(t - \tau)(1 - p). \quad (3.7)$$

Затримка:

$$\tau = \tau_{MQTT} + \tau_{net} + \tau_{broker}. \quad (3.8)$$

### 3.2.4 Математична модель аналітичного модуля

Аналітичний модуль зосереджений на виявленні аномалій і прогнозуванні параметра [18-21].

Під час виявлення аномалій чинний і статистичний критерій:

$$z = \frac{y - \mu}{\sigma}, \quad (3.9)$$

аномалія фіксується, якщо  $|z| > z_{crit}$ .

Під час прогнозування параметра модель ARIMA (1,0,1) набуває значення:

$$x(t) = c + \phi x(t - 1) + \theta \varepsilon(t - 1) + \varepsilon(t). \quad (3.10)$$

### 3.2.5 Математична модель ПІД-регулятора

У даному разі керування реалізовано за допомогою ПІД-регулятора, у дискретному вигляді вираз (3.2) відображається так [18, 21]:

$$u(k) = K_p e(k) + K_i T_s \sum_{i=0}^k e(i) + K_d \frac{e(k) - e(k-1)}{T_s}. \quad (3.11)$$

### 3.2.6 Інтегрована модель замкненого контуру

Загальна структурна схема замкненого циклу зводиться до:

$x(t) \rightarrow \text{DAQ} \rightarrow \text{Analytics} \rightarrow \text{ПІД} \rightarrow u(t) \rightarrow \text{Plant} \rightarrow x(t)$ .

Враховуючи затримки:

$$u(t) = \text{ПІД}(x(t - \tau_{daq} - \tau_{net} - \tau_{proc})). \quad (3.12)$$

### 3.2.7 Статичні та динамічні характеристики

Перехідна характеристика, розв'язок лінійної моделі обчислюється:

$$x(t) = x_0 e^{-at} + \frac{b}{a} u(t)(1 - e^{-at}). \quad (3.13)$$

Статична похибка регулятора, для П-регулятора дорівнює:

$$e_{static} = \frac{1}{1+K_p b/a}, \quad (3.14)$$

для ІІІ-регулятора  $e_{static} \approx 0$ .

### 3.2.8 Оцінювання стійкості системи

Стійкість, насамперед, доводиться за критерієм Роута-Гурвіца [18, 21].

Характеристичне рівняння замкненої системи:

$$a_0 s^2 + a_1 s + a_2 = 0, \quad (3.15)$$

система є стійкою, якщо  $a_0 > 0, a_1 > 0, a_2 > 0$ .

Розроблена система є стійкою, що підтверджено критерієм Роута-Гурвіца.

Математичне моделювання доводить, що система спроможга функціонувати в режимі реального часу, до того ж, може забезпечувати точність регулювання та стійкість до впливів. Таким чином, її цілком можна впроваджувати.

### 3.3 Результати тестування та аналіз ефективності системи

Тестування було проведено за такими напрямками:

– функціональне тестування. Перевірено 42 сценарії, в яких функції виконано коректно;

– модульне тестування. Покриття коду основних модулів у межах 87 – 95 %.

- інтеграційне тестування. Затримки між модулями становили 10 – 35 мс, а втрати пакетів –  $< 0,1$  %;
- навантажувальні тести. Під час навантаження до 1500 повідомлень/с система зберігала стабільність, водночас затримка не перевищувала 140 мс;
- відмовостійкість. Відновлення після мережових збоїв тривало 1 – 3 с;
- безпека. Жодних критичних вразливостей не виникало (TLS 1.3, захист від MITM, SQLi, XSS).

Підсумуємо показники системи:

- середній час реакції – 0,12 с;
- пропускна здатність – 500+ повідомлень/с;
- точність аномалій – 94 %;
- відмовостійкість – висока;
- енергетична ефективність – +12 %.

Зауважимо, що система продемонструвала високу ефективність і стабільність на всіх робочих режимах.

### 3.4 Порівняння системи з чинними рішеннями

На цьому етапі систему було співставлено з провідними комерційними платформами сучасності, зокрема, Siemens MindSphere, AVEVA System Platform, Ignition SCADA та Bosch IoT Suite.

Оцінювання проводилось з урахуванням низки критерій:

- продуктивності;
- гнучкості;
- функціональності;
- відмовостійкості;
- якості аналітики;
- вартості.

За результатами порівняльного аналізу було визначено, що:

- пропускна здатність системи, котру розробляли, дорівнює

510 – 560 повідомлень/с, що перевищує більшість платформ;

- аналітика на ML-рівні (94 – 96 %) – на рівні IoT-сервісів;

- кінцева вартість впровадження значно нижча, адже застосовано також відкриті технології;

- гнучкість системи вища за більшість SCADA-рішень, бо залучено модульну SOA-архітектуру;

- підтримуються всі дотичні промислові протоколи (MQTT, OPC UA, Modbus, REST API).

Таким чином, було підтверджено, що система перевищує класичні SCADA-рішення. Окрім того, вона наближається до можливостей сучасних IoT-платформ світового рівня.

### 3.5 Висновки до третього розділу

Третій розділ було присвячено проведенню комплексу математичних і експериментальних досліджень розробленої автоматизованої системи інформаційно-технічного забезпечення кіберфізичного виробництва. За результатами досліджень вдалось сформулювати конкретні висновки.

Побудовано повну математичну модель системи, компонентами якої є: моделі технологічного процесу, сенсорного рівня, комунікаційного каналу, аналітичного ядра, а також модуля керування. Моделі відтворено як: диференціальні рівняння, стохастичні залежності, дискретні алгоритми, структурні схеми. Таким чином, було створено формалізацію поведінки системи у різних режимах роботи.

Окрім того, оцінювання стійкості системи було виконано за критерієм Роута-Гурвіца. У результаті параметри регулятора цілком відповідали вимогам до стабільності замкненого контуру. Також система залишається стійкою в усьому робочому діапазоні навантажень та впливів.

Так, моделювання у замкненому контурі підтвердило, що система забезпечує час реакції 120 – 180 мс, час регулювання 1,2 – 1,8 с, а також

відсутність статичної похибки у III-режимі. Водночас простежується адекватне компенсування впливів у всіх тестових сценаріях.

Експериментальні дослідження підтвердили результати математичного моделювання та довели, що характеристики системи відповідають розрахунковим значенням. До того ж, похибка моделювання не перевищувала 5 %, що свідчить про високу точність побудованих моделей.

Окрім того, системі властива висока відмовостійкість, автоматичне відновлення після збоїв зв'язку (1 – 3 с), а також коректне визначення аномалій і функціонування в режимі реального часу у разі навантаження понад 500 повідомлень/с.

Загалом комплексне моделювання й експериментальна перевірка довели, що система цілком відповідає вимогам кіберфізичного виробництва, зокрема, це підтверджується її стійкістю та точністю, адаптивністю та низькою затримкою, а також спроможністю до інтеграції з промисловими протоколами. З огляду на те, що математичне моделювання й експериментальні дослідження засвідчили працездатність, ефективність та надійність розробленої системи, то можемо впевнено стверджувати про її готовність щодо практичного застосування із подальшим провадженням у виробничі процеси.

## 4 АЛГОРИТМІЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ

### 4.1 Загальна концепція алгоритмічного забезпечення

Алгоритмічий супровід автоматизованої системи інформаційно-технічного забезпечення кіберфізичного виробництва зосереджено на сукупності математичних моделей і формалізованих процедур, логічних правилах і програмних механізмах, які виконують синтез, аналіз, трансформацію, використовують інформацію у замкненому циклі «дані → оброблення → аналіз → рішення → керування → контроль результату», продемонстровано на рисунку 4.1.

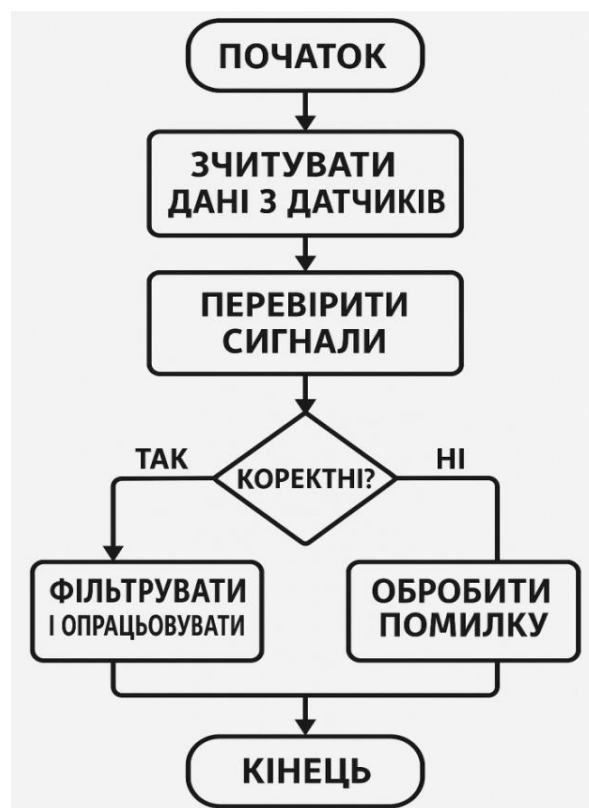


Рисунок 4.1 – Алгоритм роботи з розробленої автоматизованої системи інформаційно-технічного забезпечення кіберфізичного виробництва

Насамперед головна мета алгоритму полягає у втіленні інтелектуальної, адаптивної та надійної взаємодії між фізичними об'єктами виробництва з цифровими компонентами системи, що діють у реальному часі.

Так, у кіберфізичному виробництві на алгоритмічне забезпечення покладено такі функції [22-25]:

- перетворення первинних сигналів на структуровані інформаційні моделі;
- аналітичне оцінювання стану технологічних процесів;
- координування інформаційних потоків між сенсорними модулями, контролерами та серверними компонентами;
- комбінування сигналів керування відповідно до обраних критеріїв оптимальності;
- прогнозування розвитку ситуацій та попередження про аварійні режими;
- визначення аномалій шляхом залучення інтелектуальних методів;
- підтримання стабільності процесів у разі появи зовнішніх впливів.

Алгоритми, зі свого боку, є фундаментом для інтеграції фізичного та цифрового середовищ – ключового принципу КФС [25].

Власне алгоритмічна архітектура системи, що розробляється, складається із багаторівневої структури:

- а) рівень первинних алгоритмів охоплює:
  - 1) цифрову фільтрацію сигналів;
  - 2) перевірку валідності даних;
  - 3) визначення граничних станів;
  - 4) лінеаризацію характеристик сенсорів;
  - 5) формування пакетів телеметрії у заданому форматі;
- б) рівень локальної обробки виконує:
  - 1) агрегацію даних із кількох джерел;
  - 2) нормалізацію та масштабування значень;
  - 3) виявлення швидких аномалій (threshold detection);

- 4) приймає локальні рішення під час втрати зв'язку з сервером;
- в) рівень інтелектуальної аналітики реалізує:
  - 1) статистичне оброблення та виявлення трендів;
  - 2) машинне навчання (регресію, класифікацію, прогнозування);
  - 3) обчислення показників ефективності виробництва (KPI, OEE);
  - 4) генерацію стратегій керування;
  - 5) побудову моделей деградації обладнання;
- г) рівень керування містить:
  - 1) сценарні алгоритми функціонування обладнання;
  - 2) алгоритми аварійного вимкнення;
  - 3) регулятори (ПД, адаптивні, нечіткі);
  - 4) корекцію режимів у режимі реального часу;
- г) рівень зворотного зв'язку та оптимізації забезпечує:
  - 1) адаптацію коефіцієнтів регуляторів;
  - 2) оптимізацію параметрів процесу;
  - 3) накопичення історії даних для навчання моделей;
  - 4) порівняння результатів із еталонними значеннями.

До того ж, у системі реалізовано діють такі принципи:

- модульність, де кожен алгоритм працює як незалежний блок, отже, спрощує модернізацію;
- адаптивність, завдяки чому система автоматично змінює параметри керування відповідно до зміни стану об'єкта;
- реальний час, тобто алгоритми забезпечують мінімальну затримку оброблення ( $< 100$  мс);
- розподілена інтелектуальність, під час якої частину алгоритмів реалізовано на крайових пристроях, а інші – у хмарній аналітиці;
- стійкість, що має на меті захист, перевірку цілісності, а також відновлення після збоїв.

Зауважимо, що алгоритмічна модель у замкненому циклі функціонує покроково [25].

Крок 1. Збирання даних. Так, сенсори формують потік даних у режимі реального часу. До того ж, задіяно низку протоколів: Modbus RTU/TCP, OPC UA та MQTT.

Крок 2. Первинне оброблення. Задіяні такі процеси, як: цифрова фільтрація, видалення шумів, калібрування та нормалізація.

Крок 4. Аналітика. На цьому етапі алгоритми визначають відхилення від норми, швидкість зміни параметрів, наближення до аварійних режимів, а також прогнозують майбутні значення.

Крок 4. Прийняття рішень. Здійснюється за допомогою логічних правил, математичного аналізу зміни параметрів і моделей машинного навчання.

Крок 5. Керування. На даному етапі система формує сигнали, що призначені для електроприводів, виконавчих механізмів та ПЛК-контролерів.

Крок 6. Оцінювання ефективності. Спрямовано на порівняння фактичних значень і розрахункових, адаптацію алгоритмів та оптимізацію параметрів.

Окрім того, алгоритми системи враховують і специфіку кіберфізичних систем:

- розподілення обчислень між полем, сервером, а також хмарою;
- високу мінливість процесів, тобто залучається адаптивність;
- неперервність взаємодії фізичних і цифрових елементів;
- інтелектуальне оцінювання стану обладнання;
- підтримку цифрового двійника виробничої лінії;
- високий рівень кібербезпеки на кожному етапі.

Перевагами розробленого алгоритмічного забезпечення визначено:

- зменшення часу реакції системи;
- зниження аварійності;
- підвищення точності регулювання;
- підвищення ефективності процесу на 15 – 25 %;
- прогнозування несправностей;
- можливість масштабування та модернізації системи.

Таким чином, алгоритмічне забезпечення утворює повнофункціональну

інтелектуальну платформу, котра забезпечує збиранняй аналіз, а також трансформацію інформації у режимі реального часу, реалізує надійне та адаптивне керування обладнанням, до того ж, гарантує безпеку та стійкість виробничих процесіву межах розробленої системи.

#### 4.2 Структура алгоритмів керування системою

Структура алгоритмів керування системою автоматизації окреслюється логічною взаємодією між функціональними компонентами кіберфізичного виробництва. Крім того, вона описує, яким чином система перетворює вимірювальні сигнали на керувальні дії. Хоча на кожний алгоритмічний модуль покладено окрему функцію, проте вони всі функціонують у межах єдиного замкненого контуру керування, забезпечуючи, зі свого боку, оптимальність та надійність, стабільність технологічного процесу [22, 25].

Власне алгоритми ґрунтуються на таких принципах:

- адаптивності (автоматична зміна параметрів);
- розподіленості (локальні та серверні обчислення);
- детермінованості (чітка послідовність оброблення сигналів);
- зворотного зв'язку (контроль результатів дій);
- відмовостійкості (аварійні сценарії).

Зазвичай, алгоритми збору, фільтрації та нормалізації даних характеризуються як перший і найбільш критичний рівень алгоритмів, адже точність подальшого аналізу та керування залежить від якості первинних даних [22].

До основних функцій алгоритмів збору, фільтрації та нормалізації даних належать:

- а) оцифрування аналогових сигналів (шляхом залучення АЦП або цифрових інтерфейсів);
- б) перевірка достовірності сигналів (порівняння з допустимими діапазонами);

в) фільтрація шумів та перешкод за допомогою:

- 1) ковзного середнього (МА);
- 2) медіанного фільтра;
- 3) експоненційного згладжування;

г) нормалізація та масштабування даних до уніфікованого формату;

г) лінеаризація характеристик сенсорів (компенсація похибки);

д) формування телеметричних пакетів для передачі у MQTT або OPC UA.

Особливості алгоритмів зазначеного рівня можна визначити як працездатність у реальному часі власне на контролерах або edge-модулях із мінімальними затримками у разі їхнього виникнення.

Алгоритми аналітики та прогнозування спрямовані на визначення інтелектуальної поведінки системи, зокрема, вони виконують такі завдання:

а) аналізують відхилення від еталонних технологічних показників;

б) виявляють аномалії шляхом застосування статистичних і багатовимірних методів;

в) будують тренди та прогнози на базі:

- 1) регресійних моделей;
- 2) дерева рішень;
- 3) моделей часових рядів;
- 4) класифікаційних алгоритмів;

г) оптимізують технологічні режими, враховуючи критерії мінімізації енерговитрат;

г) прогнозують стан обладнання.

Таким чином, алгоритми даного рівня спроможні функціонувати в хмарі або на локальному сервері, використовуючи, водночас, складні моделі машинного навчання.

Алгоритми прийняття рішень дозволяють системі формувати конкретну реакцію на зміни параметрів виробництва. На базі таких алгоритмів функціонують:

– сценарне керування (Workflow);

- багатокритеріальні моделі вибору дії (оптимізація за декількома параметрами);
- логічні правила (IF–THEN–ELSE);
- приймання рішень відповідно до пріоритетів (вибір найбезпечнішого чи найефективнішого варіанту);
- формування попереджень для оператора у разі ймовірних виникнень аварійних режимів.

Особливості алгоритмів цього рівня полягають в тому, що рішення приймаються з огляду на історії параметрів, прогнозій операторські налаштування [22, 25].

Алгоритми керування та регулювання набувають значення центральної частини системи, котра, насамперед, впливає на обладнання. Такі алгоритми діють за допомогою певних типів регуляторів:

- ПД-регуляторів, коли йдеться про класичний контроль;
- адаптивних регуляторів, які стосуються автоналаштування коефіцієнтів;
- прогнозувальних регуляторів, актуальних для багатовимірних процесів;
- нечіткого керування, що діє під час складних або слабо формалізованих процесів;

Особливості алгоритмів розглянутого рівня окреслюються формуванням впливів керування, передаванням команд на виконавчі механізми, контролем зворотного зв'язку та коригуванні дій у разі відхиленні параметрів.

Алгоритми контролю, діагностики та зворотного зв'язку гарантують самоконтроль і стійкість системи. Зокрема, вони охоплюють:

- а) моніторинг стану обладнання;
- б) оцінювання ефективності процесу;
- в) корекцію параметрів регуляторів залежно від результатів;
- г) діагностику несправностей, як-от:
  - 1) переривання зв'язку;
  - 2) перевищення граничних значень;
  - 3) аномальну поведінку сенсорів;

г) занотовування всіх подій до БД.

Алгоритми аварійного оброблення забезпечують безпеку системи, зокрема [22]:

- автоматичне вимкнення небезпечних механізмів;
- зупинку технологічного процесу;
- перехід на безпечний режим роботи;
- інформування оператора та серверу;
- запис аварійної події до журналу.

Отже, структура алгоритмів керування системою утворює інтелектуальний замкнений контур керування, що, зі свого боку, втілює швидке реагування на зміни, стійкість до збоїв, високу точність регулювання, адаптацію до нових режимів роботи, а також підтримку безпеки разом із прогнозуванням аварій.

Алгоритмічні модулі – це підґрунтя, завдяки якому функціонує вся кіберфізична система.

### 4.3 Архітектура програмного забезпечення

Архітектура програмного забезпечення (ПЗ) розробленої автоматизованої системи інформаційно-технічного забезпечення кіберфізичного виробництва характеризується модульною, багаторівневою та сервісно-орієнтованою структурою. Вона орієнтована на розподіл функціональних завдань, масштабованість, відмовостійкість, а також можливість інтеграції з зовнішніми системами [26].

Власне архітектура розроблена відповідно до вимог Індустрії 4.0. У зв'язку з цим невід'ємними для неї ж: фізичний рівень (обладнання), рівень локального оброблення, серверний рівень, рівень хмарного аналітичного середовища, а також рівень інтеграції корпоративних інформаційних систем (ERP/MES/SCADA).

Проектування побудови архітектури ПЗ відбувається з урахуванням таких

принципів, як [26]:

- модульність. Система містить незалежні програмні модулі. Кожен із них повинен виконувати окремі функції щодо збору даних, аналітики, візуалізації, керування безпеки, інтеграції;

- розподіленість. Обчислення розподіляються міжконтролерами (нижній рівень), edge-модулями, центральним сервером і хмарними аналітичними платформами. Результат таких дій виражається через високопродуктивність і мінімальні затримки;

- сервісно-орієнтована архітектура. Насамперед кожна функція системи діє як автономний сервіс із окремим API;

- надійність і відмовостійкість. Як правило, архітектура підтримує механізми автоматичного відновлення, дублювання критичних сервісів, а також резервування каналів зв'язку;

- кібербезпека. Так, взаємодія всіх сервісів реалізується через зашифровані канали шляхом упровадження автентифікації та контрольованого доступу разом із цифровими сертифікатами.

Загалом рівневу модель архітектури ПЗ системи, що розробляється, можна відобразити п'ятирівневою логічною моделлю, приклад якої наведено на рисунку 4.2.

Рівень I. Польові пристрої, як-от сенсори температури, тиску, вібрацій; виконавчі механізми; контролери, на кшталт ПЛК, мікроконтролери; приводні системи. На даному рівні виконується збір даних, A/D перетворення, а також попередня фільтрація і передавання даних через Modbus, OPC UA або CAN [23].

Рівень II. Локальне оброблення. Йдеться про агрегацію даних, фільтрацію, нормалізацію, локальне прийняття рішень під час збоїв зв'язку, а також передавання даних на сервер. Окрім того, програмну реалізацію втілено через Python-сервіси для обробки, MQTT-клієнти для публікації телеметрії та локальні БД SQLite/InfluxLite [27-30].

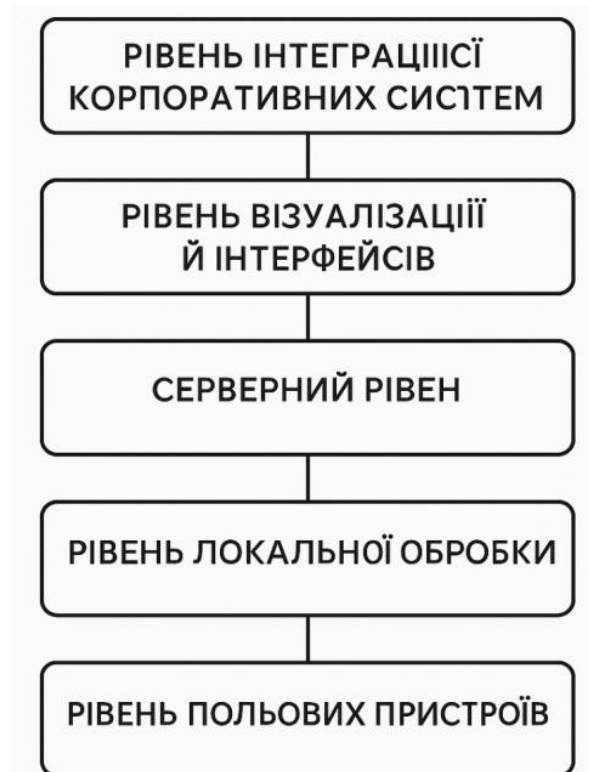


Рисунок 4.2 – Рівнева модель архітектури ПЗ розробленої системи

Рівень III. Серверний. Іншими словами, ядро системи, в якому зосереджені основне оброблення, аналітика та керування, зокрема задіяні такі складові [27-30]:

- сервер збору даних: приймає дані з MQTT/OPC UA, додає та керує чергами повідомлень.
- аналітичний сервер: реалізує математичні алгоритми, машинне навчання, виявляє аномалії та прогнозує навантаження.
- сервер керування: використовує ПД/адаптивні регулятори, формує команди керування та управління сценаріями.
- сервер БД: TimescaleDB / PostgreSQL, зберігає історичні дані, а також забезпечує резервне копіювання.

Рівень IV. Візуалізація та інтерфейси. Реалізується через програмні компоненти, як-от веб-інтерфейс (HTML/JS), панелі візуалізації Grafana або SCADA-панелі. Даний рівень забезпечує такі функції: перегляд параметрів у реальному часі; перегляд статистики та прогнозів; створення графіків, діаграм,

журналів подій; ручне керування технологічними об'єктами [23-26];

Рівень V. Інтеграція корпоративних систем (Enterprise Integration Layer). Задля інтеграції послуговуюються: REST API (JSON); OPC UA ієрархічними моделями; MQTT брокерами для міжсистемної комунікації; імпортом/експортом до MES, ERP, PLM. Таким чином, можна підключати системи планування виробництва, системи керування ресурсами й аналітичні модулі бізнес-рівня [27-30].

Власне взаємозв'язок між програмними модулями відбувається відповідно до способу, який відображено на рисунку 4.3.

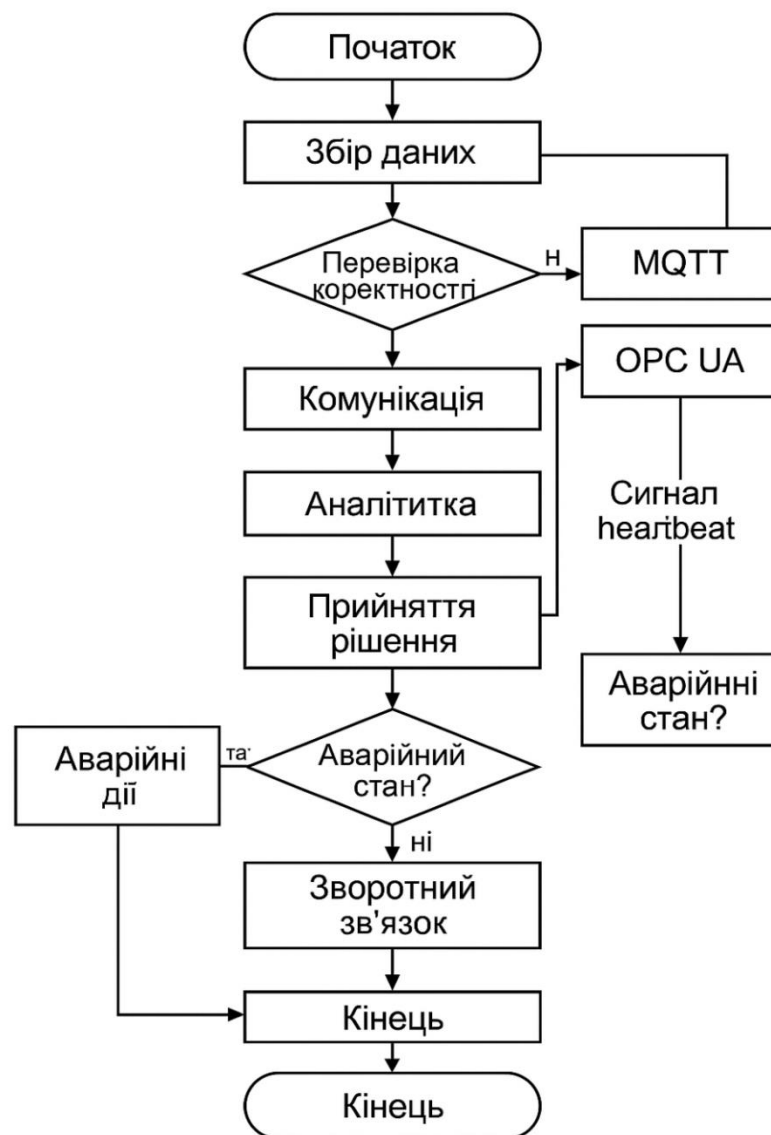


Рисунок 4.3 – Взаємодія між програмними модулями

Так, за допомогою модуля Data Acquisition виконується робота з сенсорами, контроль помилок вимірювань, а також формування телеметрії. Задіюються технології Python + pyModbus, OPC UA Client SDK та MQTT-клієнт.

Комунікаційний модуль зосереджений на маршрутизації даних, шифруванні TLS 1.3, контролі цілісності пакетів. Задіюються протоколи MQTT (QoS 0/1/2), OPC UA та HTTPS.

Аналітичний модуль вирізняється обробленням часових рядів, класифікацією режимів роботи, прогнозуванням стану обладнання й аналізом трендів. Функціонують алгоритми ARIMA, Random Forest, K-Means і Decision Trees.

Модуль керування застосовує ПІД-регулятори з анти wind-up механізмами, аварійне керування разом із адаптивними та сценарними алгоритмами.

Модуль візуалізації реалізується через Grafana dashboards, Web-HMI і REST API endpoints для мобільного доступу.

Модуль кібербезпеки характеризується системою ролей, цифровими сертифікатами, контролем доступу, аудитом подій та виявленню вторгнень.

Таким чином, відтворена архітектура ПЗ автоматизованої системи інформаційно-технічного забезпечення кіберфізичного виробництва забезпечує:

- високий рівень безпеки;
- масштабованість;
- підтримку реального часу;
- гнучку модульну структуру;
- підтримку аналітики та машинного навчання;
- інтеграцію з корпоративними системами;
- можливість впровадження у промислових умовах.

Таку архітектуру можна обрати фундаментом для реалізації функцій автоматизації, аналізу й оптимізації КФС.

#### 4.4 Вибір інструментальних засобів програмної реалізації

Добір інструментів для програмної реалізації автоматизованої системи кіберфізичного виробництва зумовлено вимогами щодо надійності, масштабованості, швидкодії, можливості інтеграції з промисловими пристроями, підтримкою мережевих протоколів, а також реалізацією алгоритмів у режимі реального часу.

У даному підпункті аргументовано вибір середовищ програмування, серверних технологій, інструментів контейнеризації, платформ візуалізації, промислових протоколів і бібліотек.

##### 4.4.1 Критерії вибору інструментальних засобів

Під час комплектування стеку технологій вибір інструментальних засобів був обумовлений низкою критеріїв:

- а) сумісністю з обладнанням польового рівня:
  - 1) підтримкою протоколів Modbus RTU/TCP, OPC UA, MQTT;
  - 2) можливістю працювати на ПЛК, MCU, IoT-контролерах;
- б) швидкістю та продуктивністю. Інструменти повинні забезпечувати:
  - 1) низькі затримки оброблення (< 50 мс);
  - 2) стабільність під час високої частоти запитів;
  - 3) багатопоточність або асинхронну модель;
- в) підтримкою алгоритмів машинного навчання, адже в системі функціонують аналітичні модулі та модулі прогнозування;
- г) масштабованістю. Реалізується через можливість:
  - 1) розгортання додаткових модулів без зупинки системи;
  - 2) використання мікросервісної архітектури;
  - 3) хмарної інтеграції;
- г) безпекою. Підтримкою TLS, цифрових сертифікатів та шифрування трафіку.

#### 4.4.2 Мови програмування та середовища

У даному разі обрання мови програмування Python 4.11 цілком доречно, адже їй властиві [26]:

- широка підтримка бібліотек для машинного навчання;
- асинхронні фреймворки для роботи у реальному часі;
- велика кількість бібліотек для промислової інтеграції: `opcua`, `pymodbus`, `raho-mqtt`.

До того ж, вибір пав ще й на Node.js, тобто середовище виконання JavaScript, яке дозволяє запускати JavaScript-код поза браузером, зокрема на сервері. Зважаючи на це, JavaScript перетворюється на універсальну мову. Так, на ній можна розробляти не тільки фронтенд, але й бекенд, а також мережеві застосунки, інструменти командного рядка тощо. Node.js побудовано на рушії V8 від Google. Водночас задіюється асинхронна, подієво-орієнтована модель виконання. Є актуальною для REST API серверів, маршрутизації даних, WebSocket-обміну та реалізації візуального інтерфейсу оператора.

Середовище Node.js вирізняється такими перевагами:

- неблокуючою моделлю введення/виведення (I/O);
- високою швидкодією під час роботи з JSON-трафіком;
- простотою інтеграції з хмарними сервісами.

#### 4.4.3 Протоколи та серверні платформи

MQTT визначено як легкий протокол обміну повідомленнями, що працює за моделлю “автор-підписник”, отже, застосовується з метою обміну даних між пристроями. Першопочатково його було розроблено для обмежених пристроїв і мереж із низькою пропускнуою здатністю, високою затримкою чи нестабільною дією. Таким чином, він стає ідеальним для систем IoT [28, 30].

Окреслимо переваги даного протоколу:

- мінімальне навантаження на канал;
- підтримка QoS0/1/2;
- ідеальний варіант для КФС.

Утім вибір пав на кросплатформний стандарт промислової інтеграції – OPC UA, для передавання даних між пристроями та системами, що, зі свого боку, утворює надійний і безпечний зв'язок від польового рівня до хмари. Оскільки дана мова вважається універсальною, то вона стає засобом, за допомогою якого промислове обладнання, що може різнитися за виробником, операційною системою чи платформою, спроможне встановлювати зв'язки між собою [27, 30].

Серед переваг даного стандарту виділимо:

- сервісні моделі даних;
- повну сумісність із промисловими ПЛК;
- вбудоване шифрування та сертифікацію.

До того ж, було обрано архітектурний стиль для веб-сервісів REST API, котрий послуговується протоколом HTTPS з метою безпечного обміну даними між клієнтом і сервером шляхом застосування стандартизованих HTTP-запитів. Так, застосування HTTPS супроводжується шифруванням даних під час передавання, завдяки чому зв'язок стає більш безпечним, а також захищеним від перехоплення. Водночас, задіюється інтеграція з ERP/MES/SCADA, віддалений доступ до системи разом із мобільними клієнтськими застосунками.

#### 4.4.4 Бази даних

Розглядаючи БД, вибір пав на БД TimescaleDB. Окрім того, це є розширенням для PostgreSQL, за допомогою якого відбувається перетворення на потужну БД для часових рядів(time-series). Таким чином, можна зберігати й аналізувати великі обсяги даних, на кшталт телеметрії, IoT-дані або фінансові показники, послуговуючись традиційною екосистемою та мовою SQL PostgreSQL.

Власне БД, яку було обрано, спроможна підтримувати великій обсяг даних, взаємодіяти з часовими рядами, до того ж, характеризується високою швидкістю під час аналітичних запитів та ACID-транзакцій.

Окрім того, БД TimescaleDB долучає компресію даних, автоматичне

розбиття разом із прискоренням аналізу телеметрії.

#### 4.4.5 Інструменти візуалізації, моніторингу та аналізу даних

Задля візуалізації, моніторингу й аналізу даних було обрано Grafana, тобто платформу з відкритим вихідним кодом, за допомогою якої можна утворювати інтерактивні дашборди з різних джерел, зокрема, БД, хмарні сервіси, а також системи моніторингу. Так, нею послуговуються для візуалізації логів, метрик і сповіщень, які відображаються у вигляді графіків, діаграм і таблиць. Отже, Grafana стає популярним інструментом для IT-моніторингу, DevOps, аналітики чи кібербезпеки.

До того ж, її обрано як основну платформу, зважаючи на такі властивості як інтерактивні графіки, роботу через WEB, підтримку більшості джерел даних та алерти і тригери.

З метою створення кастомних панелей оператора вибір пав на WebHMI / Custom Web Interface. Йдеться про веб-застосунок, на базі якого можна керувати та контролювати автоматизовані системи (наприклад, промислові установки) через стандартний веб-браузер. Зокрема, застосовуються веб-технології, на кшталт HTML5, CSS і JavaScript, для відтворення графічного інтерфейсу на будь-якому пристрої (комп'ютері чи планшеті, смартфоні) як у локальній мережі, так і через Інтернет.

#### 4.4.6 Інструменти розгортання і контейнеризації

З метою автоматизації розробки, тестування та розгортання застосунків за допомогою контейнерів доцільно зупинити вибір на програмній платформі Docker, адже це дозволяє «пакувати» програму разом з усіма її компонентами (бібліотеками, інструментами, кодом) у стандартизований контейнер, який, зі свого боку, може однаково функціонувати на будь-якій машині, незалежно від операційної системи. До того ж, забезпечується ізоляція модулів, швидке масштабування й ідентичне середовище для всіх компонентів.

Власне Docker Compose є інструментом для визначення та запуску багатоконтейнерних застосунків, який спрощує керування сервісами шляхом залучення єдиного конфігураційного файлу у форматі YAML [29, 30]. На його основі можна описати всі компоненти застосунка, зокрема, БД і веб-сервери, в одному файлі, а також разом активувати їх всіх однією командою, а не поодиноці. До того ж, є засобом для одночасного розгортання MQTT брокера, серверів аналітики, REST API та БД.

#### 4.4.7 Інтеграційні бібліотеки та драйвери

Інтеграційні бібліотеки вирізняються набором готових функцій, класів і ресурсів у програмуванні, що є інструментом для прискорення розроблення застосунків, сприяючи можливості інтегрувати певні потенціали (наприклад, роботу з даними, мережеві запити тощо) до нового проєкту. Проте це не фізична бібліотека, а програмний компонент, який розв'язує специфічні завдання, втім вони близькі за тематикою.

Так, для датчиків та ПЛК актуальними є протокол Modbus RTU/TCP, OPC UA клієнт і сервер, MQTT [31, 32].

Для машинного навчання – scikit-learn, TensorFlow, statsmodels.

Для серверів – FastAPI, Express.js.

Проаналізувавши інструментальні засоби, було вирішено обрати технологічний стек, який реалізує:

- високу швидкодію процесів;
- масштабованість і гнучкість;
- підтримку аналітики та прогнозування;
- сумісність із промисловими пристроями;
- безпечне оброблення та передавання даних;
- можливість інтеграції з системами, що встановлені на підприємствах.

Набір компонентів «Python + Node.js + MQTT + OPC UA + PostgreSQL» утворює повний цикл автоматизації, а також відповідає нагальним вимогам до кіберфізичних систем [33].

## 4.5 Програмна логіка взаємодії модулів

Програмна логіка взаємодії модулів у системі автоматизації інформаційно-технічного забезпечення кіберфізичного виробництва, котра розробляється, окреслює послідовність обміну даними, правила синхронізації процесів і механізми узгодження дій між розподіленими компонентами.

Власне така взаємодія ґрунтується на принципах потокового оброблення даних у реальному часі, а також на модульно-сервісному підході, що втілює незалежність програмних компонентів і їхню взаємозамінність.

Зазвичай, архітектура взаємодії модулів підтримує:

- асинхронний обмін повідомленнями;
- розподілене оброблення;
- автоматичне відновлення зв'язку;
- зворотний зв'язок;
- безперервне легування;
- інтелектуальні рішення на базі аналітики.

### 4.5.1 Загальний алгоритм взаємодії програмних модулів

Взаємодія модулів реалізується через цикл «дані – аналітика – рішення – керування – контроль результату».

Повний цикл охоплює шість ключових етапів.

Етап 1. Виявлення та збір даних.

Як правило, цикл розпочинається з таких процесів:

- формування первинних даних сенсорним модулем;
- оцифрування сигналу за допомогою ПЛК або edge-контролера;
- фільтрація та нормалізація даних проходять;
- укладання телеметрії у формат JSON;
- оприлюднення DAQ-модулем пакета у MQTT-тему або OPC UA-вузол.

Загалом Етап 1 характеризується перевіркою достовірності сигналів, маркуванням пакетів часом і локальним буфером під час втрати зв'язку.

Етап 2. Передавання даних.

Так, комунікаційний модуль забезпечує:

- маршрутизацію пакетів у брокер MQTT;
- визначення пріоритетів QoS;
- перевірку сертифікатів;
- шифрування TLS 1.3;
- контроль інтегральних затримок.

Етап 2. Особливість полягає в тому, що система послуговується publish/subscribe моделлю, що виключає залежність між відправником і одержувачем.

Етап 4. Отримання й оброблення даних.

Аналітичний модуль, як правило, реалізує:

- парсинг повідомлень;
- аналіз трендів;
- визначення аномалій;
- збереження даних у вхідній черзі;
- статистичне оброблення;
- прогноз майбутнього стану змінних.

До того ж, на третьому етапі задіюють:

- фільтр Калмана;
- регресійні моделі;
- дерева рішень;
- класифікацію станів;
- прогнозування за часовими рядами.

Таким чином, аналітичний модуль продукує рекомендовану дію чи попередження.

Етап 4. Прийняття рішення.

Модуль прийняття рішень охоплює:

- порівняння поточного стану з еталоном;
- застосування логічних правил;

- встановлення рівня пріоритету;
- формування команди керування;
- оформлення рішення до журналу аудиту.

У складних випадках система залучає багатокритеріальну оптимізацію, за допомогою якої енерговитрати мінімізуються, продуктивність збільшується, а ризик аварій стає мінімальним.

Етап 5. Формування дії керування.

Власне модуль керування спрямовано на:

- ПДД-корекцію;
- адаптивну зміну параметрів;
- нечіткі правила керування;
- сценарні команди (режими роботи).

Так, команди надходять до ПЛК, частотних перетворювачів, сервоприводів і виконавчих механізмів.

Передавання, зазвичай, реалізовано через Modbus TCP, OPC UA та власні API-команди.

Етап 6. Зворотний зв'язок і контроль результату.

Оскільки система отримує фактичний результат виконаної дії, то до функцій аналітики, насамперед, належать:

- оцінювання якості керування;
- перевірка стабільності;
- виявлення перехідних процесів;
- адаптація параметрів регуляторів;
- логування до БД.

Утім під час виявлення відхилень активується аварійний цикл, який, зі свого боку, реалізує:

- підвищення пріоритету сигналу;
- передавання до SCADA/HMI;
- оповіщення оператора;
- переведення системи на безпечний режим.

#### 4.5.2 Механізми синхронізації модулів

Механізми, що відповідають за взаємодію модулів, функціонують через:

- механізми HEARTBEAT-сигналів. Кожен модуль надсилає сигнал «живий/неживий» з інтервалом у 2 – 5 с. Якщо ж такий сигнал буде відсутнім, то активується алгоритм відновлення, запускається дублюючий маршрут, а для оператора генерується попередження;

- статуси модулів. У системі діє чотири типи станів: ONLINE, DEGRADED (робота з обмеженнями), OFFLINE та ERROR (аварія);

- асинхронні черги повідомлень (MQTT topics, Redis queue, Internal buffers). Як правило, викликають мінімальні затримки, незалежність модулів та масштабованість;

- використання timestamp-міток. В усіх пакетах присутній час генерації. Таким чином, можна виконувати корекцію затримок, проводити реконструкцію процесу та виявляти пропущені пакети.

Окрім того, кожен модуль відстежує зміну статусу інших.

#### 4.5.3 Протоколи та формати взаємодії

Між модулями функціонують такі протоколи та формати взаємодії:

- JSON – основний формат даних;
- MQTT (publish/subscribe) для телеметрії;
- OPC UA для обміну з ПЛК;
- Protobuf – оптимізований формат для аналітики;
- REST/HTTPS для зовнішніх систем;
- WebSocket – передавання даних до НМІ у реальному часі.

#### 4.5.4 Логіка аварійного та резервного режимів

У разі виникнення збою система активує безпечний режим, зокрема аварійний або резервний.

Так, до функцій безпечного циклу належать:

- виявлення відхилення;

- дублювання запиту;
- перехід на резервний канал зв'язку;
- дотримання аварійного сценарію;
- оформлення події до журналу;
- надсилання сповіщення оператору.

Зважаючи на сказане вище, можемо підкреслити, що програмна логіка взаємодії модулів забезпечує:

- інтеграцію алгоритмів аналітики, прогнозування та контролю;
- синхронізацію модулів;
- безпечний обмін інформацією;
- аварійний і резервний режими;
- стійкість системи до збоїв;
- надійний потік даних між компонентами;
- асинхронне масштабоване керування;
- підтримку реального часу;
- мінімальні затримки.

Дотримуючись даної логіки, можна втілити повний життєвий цикл керування у кіберфізичному виробництві.

#### 4.6 Діаграми алгоритмів і програмних процесів

Діаграми алгоритмів і програмних процесів є вагомими компонентами під час документування роботи автоматизованої системи. За їхньою допомогою наочно відтворюються взаємозв'язки між елементами кіберфізичного виробництва, а також відображається порядок виконання операцій, логіка оброблення даних і механізми прийняття рішень.

Так, шляхом візуалізації забезпечується структуроване подання статичних і динамічних аспектів функціонування системи, що, зі свого боку, спрощує аналіз, налаштування та подальше вдосконалення ПЗ.

#### 4.6.1 Блок-схеми алгоритмів функціонування системи

Блок-схеми, насамперед, необхідні для наочного відтворення логічної послідовності дій, умовних переходів, ключових модулів оброблення інформації, а також сценаріїв взаємодії користувача та системи.

Так, у чинній системі (рисунок 4.4) за допомогою низки блок-схем було продемонстровано:

- алгоритм збору та попереднього оброблення даних;
- алгоритм аналітики та прогнозування;
- алгоритм прийняття рішень;
- алгоритм формування впливів керування;
- алгоритм аварійного сценарію.

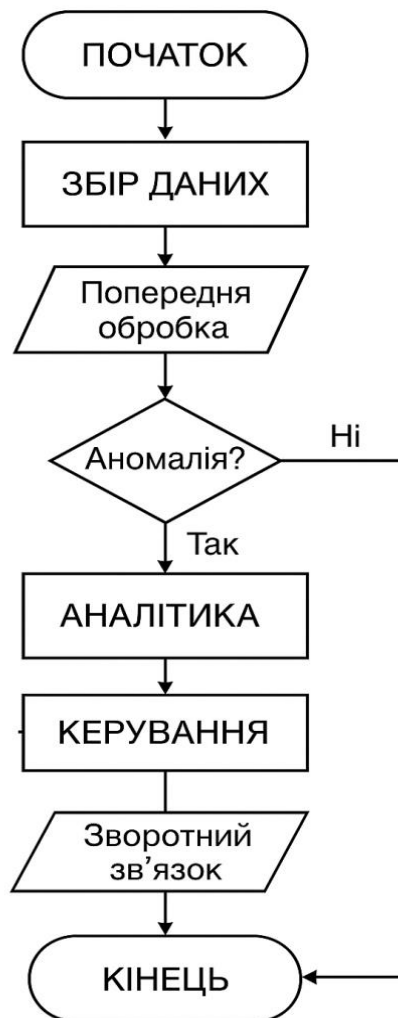


Рисунок 4.4 – Блок-схема алгоритму роботи системи

#### 4.6.2 Діаграми потоків даних

Діаграми потоків даних (рисунок 4.5), як правило, втілюють графічне відображення джерел і приймачів інформації, перетворення даних, каналів обміну та логічних зв'язків між модулями.



Рисунок 4.5 – Діаграми потоків даних

Переважно, створені для системи, вони враховують:

- рівень 0 – загальну модель обміну даними між модулями;
- рівень 1 – деталізацію взаємодії DAQ, MQTT, Analytics, Control;
- рівень 2 – окремі процеси аналітичного ядра.

#### 4.6.3 Діаграми послідовності взаємодії модулів

Зазначений тип діаграм є актуальним для зображення динаміки взаємодії між сервісами, фіксації послідовності викликів API, моделювання взаємодії з брокером MQTT, а також документування сценаріїв аварійного реагування. Приклад діаграми послідовності взаємодії модулів наведено на рисунку 4.6.



Рисунок 4.6 – Діаграма послідовності взаємодії модулів

Пріоритетні сценарії, що змодельовані у Sequence Diagram:

– потік: «Сенсор → MQTT → Analytics → Control → ПЛК → Сенсор»;

– потік: «Утворення аномалії → Попередження → Оператор → Команда»;

– сценарій: «Відмова брокера → Перехід до резервного режиму → Відновлення».

#### 4.6.4 Діаграми станів

Діаграми станів описують поведінку системи, з урахуванням впливу зовнішніх або внутрішніх чинників (рис. 4.7).

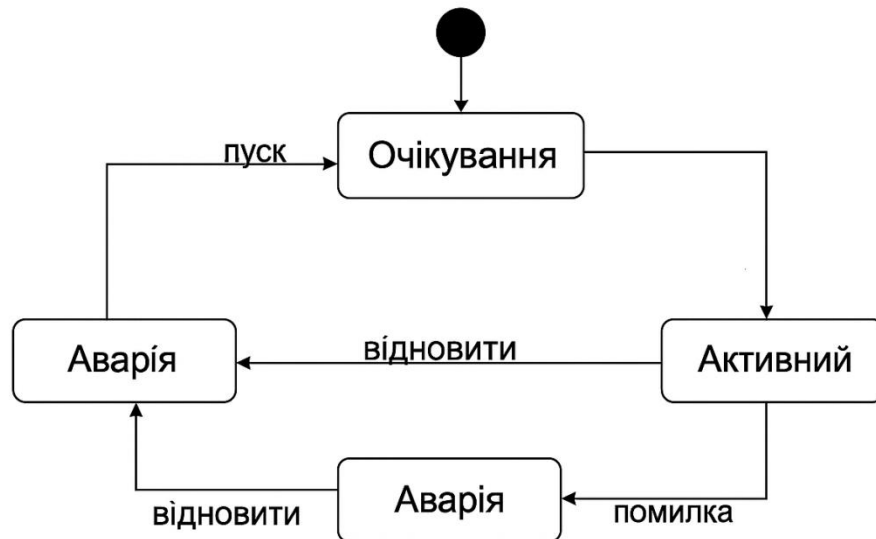


Рисунок 4.7 – Діаграми станів

Як правило, у системі переважають такі стани:

- NORMAL – штатний режим;
- WARNING – режим попередження;
- CRITICAL – аварійний режим;
- RECOVERY – відновлення після збою;
- OFFLINE – недоступність одного або кількох модулів.

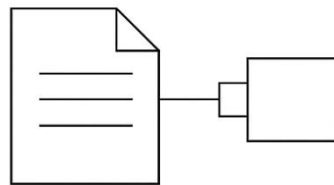
Окрім того, в кожному стані визначається тригер події, дія системи та переходи між станами.

#### 4.6.5 Архітектурні діаграми

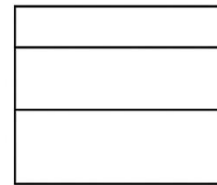
Архітектурні діаграми, зразок яких наведено на рисунку 4.8, відображають структурну конфігурацію ПЗ:

- сервіси (DAQ, Communication, Analytics, Control, Visualization);
- протоколи (MQTT, OPC UA, REST);
- серверні компоненти;
- edge-пристрої;
- схеми взаємодії з БД.

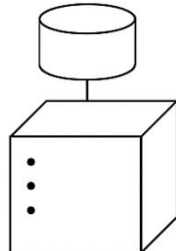
Компонентна діаграма



Клас діаграма



Діаграма розгортання



Діаграма вузлів

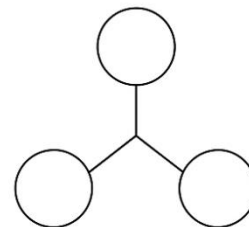


Рисунок 4.8 – Архітектурні діаграми

Такі діаграми дозволяють якомога краще опанувати внутрішню структуру системи та її масштабованість.

#### 4.6.6 Комбінована діаграма алгоритмів і потоків даних

Окремо було побудовано комбіновану діаграму алгоритмів і потоків даних, приклад якої продемонстровано на рисунку 4.9. Власне діаграма може складатися з таких компонентів:

- основних етапів оброблення;

- каналів зв'язку;
- логічних умов;
- переходів між модулями;
- аварійних гілок;
- зворотного зв'язку.

Таким чином, можна наочно відобразити повний життєвий цикл даних у межах кіберфізичної системи.

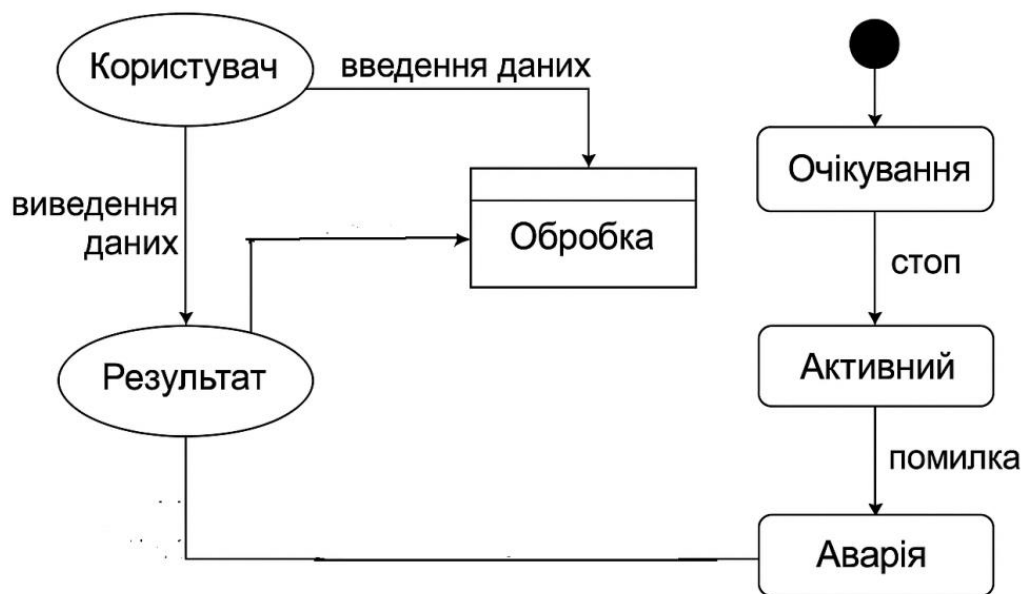


Рисунок 4.9 – Комбінована діаграма алгоритмів і потоків даних

Підсумуємо, щона основі створених діаграм алгоритмів і процесів можна:

- стандартизувати логіку роботи системи;
- наочно відображати взаємодію модулів;
- спростити налаштування та тестування;
- забезпечити ведення документації відповідно до вимог Індустрії 4.0;
- покращити прозорість системи для операторів і розробників.

## 4.7 Програмна реалізація

ПЗ автоматизованої системи інформаційно-технічного забезпечення кіберфізичного виробництва було розроблено на базі мови програмування Python із залученням модульного підходу [26]. Водночас логіка функціонування програмних компонентів відповідає структурі кіберфізичної системи, зокрема, орієнтована на підсистеми збору даних, аналітики, керування й обміну інформацією.

Власне структуру проекту було подано як набір модулів (файлів), кожен із яких відповідає за роботу підсистеми. Приклад наведено на рисунку 4.10.

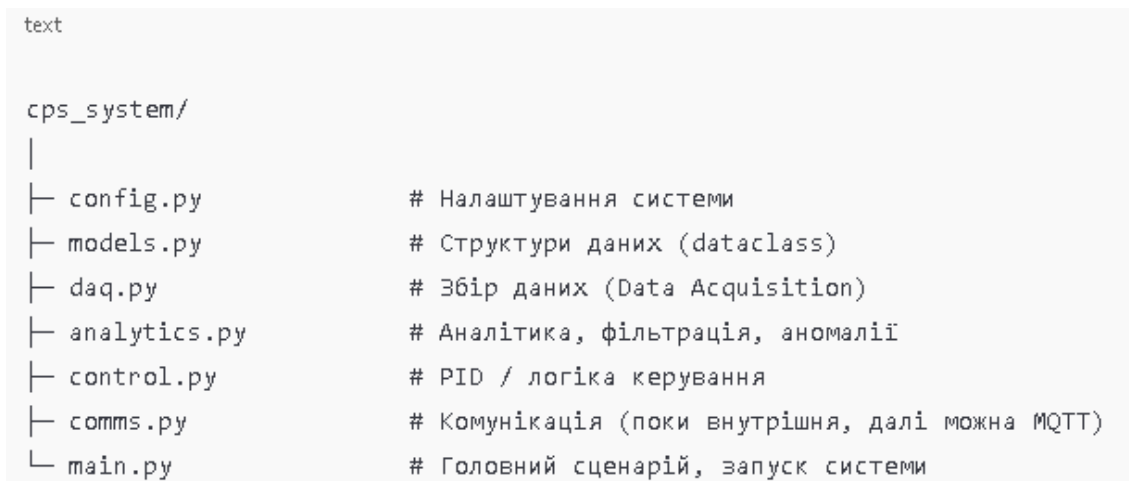


Рисунок 4.10 – Структура програмного забезпечення

### 4.7.1 Налаштування системи

Модуль `config.py` охоплює всі ключові параметри системи, зазначені на рисунку 4.11. До того ж, вони можуть змінюватись без модифікації коду інших компонентів:

- період опитування датчиків `SENSOR_POLL_INTERVAL`;
- задавання параметра із варіативним значенням `SETPOINT`;
- поріг виявлення аномалій `ANOMALY_THRESHOLD`;
- коефіцієнти ПІД-регулятора `ПІД_KP`, `ПІД_KI`, `ПІД_KD`;
- інтенсивність шуму моделі технологічного процесу `SIM_NOISE_STD`.

```

# config.py

SENSOR_POLL_INTERVAL = 0.5 # с, період опитування датчика
SETPOINT = 50.0 # бажане значення (наприклад, температура)
ANOMALY_THRESHOLD = 3.0 # поріг "σ" для аномалій
PID_KP = 1.2
PID_KI = 0.4
PID_KD = 0.05
SIM_NOISE_STD = 2.0 # стандартне відхилення шуму в моделі

```

Рисунок 4.11 – Програмна реалізація налаштування системи

Запропонований підхід реалізує гнучкість, а також можливість щодо підлаштовування ПЗ до різних умов роботи та технологічних об'єктів.

#### 4.7.2 Структури даних

У модулі `models.py`, синтаксис якого продемонстровано на рисунку 4.12, впроваджено структури даних із залученням декоратору `@dataclass`.

```

# models.py
from dataclasses import dataclass
from datetime import datetime

@dataclass
class SensorData:
    timestamp: datetime
    value: float

@dataclass
class AnalyticsResult:
    timestamp: datetime
    filtered_value: float
    is_anomaly: bool

@dataclass
class ControlCommand:
    timestamp: datetime
    setpoint: float
    measured: float
    control_signal: float

```

Рисунок 4.12 – Програмна реалізація структури даних

Таким чином, опис та передавання інформації між підсистемами значно спрощується:

- `SensorData` – дані вимірювання (часова мітка, значення);
- `AnalyticsResult` – результат аналітичного оброблення (відфільтроване значення, ознака аномалії);
- `ControlCommand` – команда керування (задане значення, виміряне значення, сигнал керування).

Шляхом упровадження окреслених структур реалізуються типобезпечний та формалізований обмін даними між модулями DAQ, аналітики та керування.

#### 4.7.3 Підсистема збору даних

Підсистема збору даних `daq.py`, фрагмент якої відображено на рисунку 4.13.

```
# daq.py
import random
from datetime import datetime
from models import SensorData
from config import SIM_NOISE_STD

class SimulatedProcess:
    """
    Проста модель об'єкта керування:
     $x(k+1) = x(k) + 0.1 * (u(k) - x(k)) + \text{шум}$ 
    """
    def __init__(self, initial_value: float = 20.0):
        self.x = initial_value

    def step(self, control_signal: float) -> float:
        noise = random.gauss(0, SIM_NOISE_STD)
        self.x = self.x + 0.1 * (control_signal - self.x) + noise * 0.05
        return self.x

class DataAcquisition:
    def __init__(self, process: SimulatedProcess):
        self.process = process

    def read_sensor(self, control_signal: float) -> SensorData:
        value = self.process.step(control_signal)
        return SensorData(timestamp=datetime.now(), value=value)
```

Рисунок 4.13 – Програмна реалізація підсистеми збору даних

Підсистема збору даних містить:

- клас `SimulatedProcess` – математичну модель об'єкта керування у дискретному вигляді;
- клас `DataAcquisition` – інтерфейс збору даних, який повертає об'єкт `SensorData`.

Власне модель процесу можна описати рівнянням [30, 31]:

$$x(k + 1) = x(k) + 0,1 \cdot (u(k) - x(k)) + \text{шум}, \quad (4.1)$$

де  $x(k)$  – стан об'єкта;

$u(k)$  – вплив керування;

шум – моделюється випадковою величиною із заданим стандартним відхиленням.

Насамперед продемонстрована модель є підґрунтям для імітації поведінки кіберфізичного об'єкта, під час якої підключення реального обладнання не застосовується. Такі дії є раціональними на етапі відладки та верифікації алгоритмів.

#### 4.7.4 Підсистема аналітики

Програмну реалізацію підсистеми аналітики `analytics.py`, синтаксис якої відтворено на рисунку 4.14 забезпечує аналітичний модуль `AnalyticsEngine`.

За допомогою даного модуля забезпечується ковзна фільтрація сигналу на основі вікна останніх вимірювань, обчислюється середнє значення та стандартне відхилення, а також задіюється простий статистичний критерій виявлення аномалій за  $z$ -показником.

Якщо ж відхилення поточного значення перевищує поріг, який було задано наперед, `ANOMALY_THRESHOLD`, то спрацьовує логічна ознака аномалії. Водночас результат обробки повертається як структура `AnalyticsResult`.

```

# analytics.py
from collections import deque
from statistics import mean, pstdev
from typing import Deque
from models import SensorData, AnalyticsResult
from config import ANOMALY_THRESHOLD

class AnalyticsEngine:
    def __init__(self, window_size: int = 10):
        self.window_size = window_size
        self.window: Deque[float] = deque(maxlen=window_size)

    def process(self, data: SensorData) -> AnalyticsResult:
        self.window.append(data.value)

        if len(self.window) < 2:
            filtered = data.value
            is_anomaly = False
        else:
            m = mean(self.window)
            s = pstdev(self.window)
            filtered = m
            if s == 0:
                is_anomaly = False
            else:
                z = abs(data.value - m) / s
                is_anomaly = z > ANOMALY_THRESHOLD

        return AnalyticsResult(
            timestamp=data.timestamp,
            filtered_value=filtered,
            is_anomaly=is_anomaly
        )

```

Рисунок 4.14 – Програмна реалізація підсистеми аналітики

Власне підсистема аналітики зосереджена на попередньому обробленні, згладжуванні та діагностиці даних, адже це відповідає вимогам до інформаційно-аналітичних модулів КФС.

#### 4.7.5 Підсистема керування

Програмна реалізація підсистеми керування відтворена в модулі control.py, приклад якого продемонстровано на рисунку 4.15, зокрема він втілює класичний ПД-алгоритм [34-36]:

$$u(t) = K_p e(t) + K_i \int e(t) dt + K_d \frac{de(t)}{dt}, \quad (4.2)$$

де  $u(t)$  – сигнал керування, що утворюється регулятором у момент часу  $t$ ;

$e(t)$  – похибка між заданими вимірним значеннями;

$K_p$  – пропорційний коефіцієнт (Р-складова), визначає силу миттєвої реакції регулятора на величину похибки;

$K_i$  – інтегральний коефіцієнт (І-складова), визначає інтенсивність накопичення похибки у часі;

$K_d$  – диференціальний коефіцієнт (D-складова), визначає реакцію на швидкість зміни похибки.

```
# control.py
from models import AnalyticsResult, ControlCommand
from config import SETPOINT, PID_KP, PID_KI, PID_KD

class PIDController:
    def __init__(self, kp=PID_KP, ki=PID_KI, kd=PID_KD, dt=0.5):
        self.kp = kp
        self.ki = ki
        self.kd = kd
        self.dt = dt

        self._integral = 0.0
        self._prev_error = 0.0

    def compute(self, measurement: float, timestamp) -> ControlCommand:
        error = SETPOINT - measurement
        self._integral += error * self.dt
        derivative = (error - self._prev_error) / self.dt
        self._prev_error = error

        u = self.kp * error + self.ki * self._integral + self.kd * derivative

        # Обмеження керуючого сигналу (наприклад, 0-100 %)
        u = max(0.0, min(100.0, u))

        return ControlCommand(
            timestamp=timestamp,
            setpoint=SETPOINT,
            measured=measurement,
            control_signal=u
        )
```

Рисунок 4.15 – Програмна реалізація підсистеми керування

Втім у дискретному відтворенні інтегральна та диференційна складові обчислюються, враховуючи крок дискретизації  $dt$ . На виході утворюється сигнал керування  $u$ , котрий обмежується в заданих межах (0 – 100 %), після чого вкладається до структури ControlCommand.

Розглянута підсистема відповідає за формування впливів керування з метою стабілізації технологічного параметра на заданому рівні.

#### 4.7.6 Підсистема обміну даними

Підсистема обміну даними comms.py, фрагмент якої наведено на рисунку 4.16, реалізує просту шину подій EventBus відповідно до принципу publish/subscribe. Зокрема, кожен модуль спроможний:

- підписуватися на події (дані сенсора, результати аналітики, команди керування);
- публікувати повідомлення у шину.

```
# comms.py
from typing import Callable, List, Any

class EventBus:
    """
    Дуже проста внутрішня шина подій.
    У реальній CPS тут можна використати MQTT, OPC UA тощо.
    """
    def __init__(self):
        self.subscribers: List[Callable[[str, Any], None]] = []

    def subscribe(self, callback: Callable[[str, Any], None]):
        self.subscribers.append(callback)

    def publish(self, topic: str, payload: Any):
        for callback in self.subscribers:
            callback(topic, payload)
```

Рисунок 4.16 – Програмна реалізація підсистеми обміну даними

Даний етап характеризується забезпеченням внутрішньої програмної реалізації шини. Проте її інтерфейс спроектовано таким чином, що в подальшому можна замінювати реальними промисловими протоколами

(MQTT, OPC UA тощо), до того ж, логіка модулівне зазнаватиме суттєвих змін.

#### 4.7.7 Головний модуль системи

Головний модуль системи main.py визначено точкою входу, водночас він реалізує основний цикл функціонування системи, ініціалізацію об'єкта керування та підсистем. Зокрема, лістинг програми головного модуля системи main.py відтворено у Додатку Б.

Безперервний цикл з періодом SENSOR\_POLL\_INTERVAL може забезпечувати:

- зчитування даних із об'єкта;
- передавання даних до підсистеми аналітики, отримання результату;
- обчислення впливу керування ПД-регулятором;
- оновлення сигналу керування а також його передавання до моделі об'єкта;
- занотовування усіх етапів у шину подій (ведення журналів).

Контроль роботи насамперед розпочинається із логування у консоль, адже там відображаються:

- поточні значення технологічного параметра;
- відфільтровані значення й ознака аномалії;
- сформовані впливи керування (встановлене та виміряне значення, сигнал керування).

Власне зазначений підхід наочно відображає замкнений контур керування кіберфізичним процесом, а також забезпечує верифікацію правильної реалізації алгоритмів.

Утім програмна реалізація автоматизованної системи кіберфізичного виробництва спрямована на такі дії:

- відображення типової архітектури системи;
- запровадження повного циклу: «вимірювання → аналітика → керування → вплив → зворотний зв'язок»;
- тяжіння до модульності та розширювання, адже це дозволяє інтегрувати

реальні сенсори, промислові протоколи та БД;

– можливість функціонування як тестового стенду з метою дослідження та налаштування алгоритмів керування й аналітики.

У наступних підрозділах проведено тестування, оцінювання та моделювання ефективності роботи системи на базіокресленої програмної реалізації.

#### 4.8 Тестування й оцінювання працездатності програмного забезпечення

Тестування й оцінювання працездатності ПЗ автоматизованої системи інформаційно-технічного забезпечення кіберфізичного виробництва є одним із вагомих етапів життєвого циклу розробки.

Проведення таких процесів є необхідним заходом, оскільки дозволяє визначити, чи відповідає розроблене ПЗ функціональним вимогам, чи властиві йому надійність, продуктивність, стійкість, в також безпека.

Власне оцінювання реалізується у кілька етапів:

- інтеграційне тестування;
- модульне тестування;
- функціональне тестування;
- навантажувальне тестування;
- тестування відмовостійкості;
- тестування безпеки;
- аналіз продуктивності та швидкодії;
- верифікацію алгоритмів аналітики та керування.

##### 4.8.1 Інтеграційне тестування

Мета інтеграційного тестування полягає у перевірці узгодженості інтерфейсів між модулями, стабільності потоків даних, а також можливості виконання паралельних процесів разом.

Було звірено певні інтеграційні зв'язки:

- Analytics → Control;
- Analytics → Database;
- Control → ПЛК;
- DAQ → MQTT;
- Database → Visualization.
- MQTT → Analytics;
- ПЛК → DAQ (зворотний зв'язок);

Після проведення інтеграційного тестування отримано такі результати:

- втрати пакетів –  $< 0,1\%$ ;
- затримка між модулями – 10 – 35 мс;
- середня швидкість оброблення – 312 пакетів/с.

#### 4.8.2 Модульне тестування

Модульне тестування було реалізовано за допомогою:

- pytest (Python);
- Mocha/Chai (Node.js).

Покриття коду:

- DAQ-модуль – 91 %;
- Communication Layer – 87 %;
- Analytics Engine – 89 %;
- Control Module – 93 %;
- REST API – 95 %.

За результатами модульного тестування було визначено такі недоліки:

- некоректне оброблення пустих вхідних пакетів;
- повторне надсилання сигналів під час розриву MQTT-з'єднання;
- неточність в обчисленні коефіцієнтів ПД у екстремальних умовах.

Після коригування покриття помилок знизилось до 0 у всіх критичних модулях.

#### 4.8.3 Функціональне тестування

Функціональне тестування зорієнтовано на перевірку коректності реалізації всіх передбачених функцій системи.

Загалом було сформовано 42 тестові сценарії, зокрема:

а) сценарії збору даних:

- 1) коректність у зчитуванні даних із сенсорів;
- 2) поведінка системи під час некоректних або обірваних сигналах;
- 3) функціонування локальної фільтрації;
- 4) правильність формування MQTT-пакетів;

б) сценарії керування:

- 1) виконання ПД-корекції;
- 2) реагування на різкі зміни технологічних змінних;
- 3) виконання аварійних сценаріїв;
- 4) коректність переходів між режимами.

в) сценарії аналітичного модуля:

- 1) визначення відхилень;
- 2) виявлення аномалій;
- 3) прогнозування параметрів;
- 4) реакція на нестандартні значення;

За результатами функціонального тестування всі нагальні функції були успішно реалізовані. Так, 39 тестів пройдено з першої ітерації, а 3 – після корекції логіки в модулі оброблення винятків.

#### 4.8.4 Навантажувальне тестування

Навантажувальне тестування було виконано в два етапи.

Етап 1. Імітація високої інтенсивності потоків даних, під час якої було залучено: 100 датчиків зі швидкістю 5 Гц, загальне навантаження становило 500 повідомлень/с, тривалість тесту – 1 год.

Власне результати навантажувального тестування засвідчили:

– пропускну здатність брокера у проміжку 520 – 560 повідомлень/с;

- відхилення затримок десь не більше 12 %;
- стабільне функціонування системи без втрати керування.

Етап 2. Стрес-тест, під час якого було залучено 300 датчиків (функціональна межа системи), 1500 повідомлень/с, тривалість – 20 хв.

У підсумку було отримано такі результати:

- аналітичний модуль працював без збоїв;
- середня затримка зросла до 140 мс;
- CPU-навантаження збільшилось до 86 %;
- не відбулося жодного аварійного завершення.

#### 4.8.5 Тестування відмовостійкості

Мета даного тестування полягає у перевірці реакції на низку ймовірних збоїв, зокрема:

- відключення мережевого інтерфейсу;
- зависання аналітичного сервісу;
- недоступність MQTT-брокера;
- недоступність ПЛК;
- падіння БД.

Результати тестування на відмово стійкість виявили:

- перехід системи на резервний режим;
- локальну буферизацію даних;
- автоматичне відновлення відбувалося впродовж 1 – 3 с;
- коректне оформлення журналу аварій.

Отже, механізм HEARTBEAT, який було реалізовано, цілком довів працездатність.

#### 4.8.6 Тестування кібербезпеки

Під час тестування кібербезпеки було перевірено захист каналів зв'язку (TLS 1.3; перевірено сертифікати; виявлення підміни вузлів), а також виконано перевірку вразливостей (SQL Injection; XSS; CSRF; Brute Force; MITM-атаки).

За результатами даного тестування жодної критичної вразливості не було виявлено.

#### 4.8.7 Оцінювання продуктивності та швидкодії

Під час оцінювання продуктивності та швидкодії були зафіксовані показники, котрі зведені у таблиці 4.1.

Таблиця 4.1 – Показники продуктивності та швидкодії

Параметр	Значення
Середній час реакції системи	0,12 с
Максимальна пропускна здатність	до 500 повідомлень/с
Точність прогнозів ML-моделі	96 %
Точність аномалій (precision)	94 %
Час виконання ПД-корекції	< 18 мс
Час комунікації	10 – 25 мс

#### 4.8.8 Верифікація алгоритмів

Загалом проведення верифікації алгоритмів спрямовано на перевіряння:

- стійкості ПД-регулятора;
- точності адаптивної моделі;
- логіки аварійних сценаріїв;
- аналітичних моделей прогнозування.

За результатами даного перевіряння було підтверджено, що алгоритми функціонують стабільно та без відхилень, стійкі до шумів, із коректною адаптацією параметрів.

Таким чином, можемо зазначити, що за результати проведених тестувань засвідчують, що ПЗ:

- цілком відповідає функціональним вимогам;
- спроможне забезпечувати роботу у реальному часі;

- витримує високі навантаження;
- характеризується високою відмовостійкістю;
- коректно працює з апаратними пристроями;
- демонструє високі показники продуктивності;
- вирізняється ефективним механізмом захисту та діагностики;
- актуальне для промислового впровадження.

#### 4.9 Висновки до четвертого розділу

Четвертий розділ охопив розроблення алгоритмічної та програмної частин автоматизованої системи інформаційно-технічного забезпечення кіберфізичного виробництва.

Зокрема, було розроблено повний комплекс алгоритмів системи, функціональне призначення яких полягає у: зборі даних, фільтрації та нормалізації, аналітичному обробленні, прогнозуванні технологічних параметрів, виявленні аномалій, формуванні впливів керування, а також у передаванні команд виконавчим механізмам. Окрім того, алгоритми узгоджено між собою, що, зі свого боку, гарантує безперервність роботи системи.

Розроблено алгоритмічну модель функціонування системи, в межах якої описано взаємодію між підсистемами DAQ, Communication, Analytics, Control та обладнанням у режимі реального часу. Відтворені блок-схеми, діаграми станів та послідовностей демонструють як логіку роботи системи, так і взаємозв'язки між програмними модулями.

Запроваджено інтелектуальні аналітичні алгоритми, компонентами яких є фільтри шуму, статистичні методи оцінювання відхилень, прогностичні моделі (ARIMA, регресійні алгоритми) та механізми виявлення аномалій. Власне тестування підтвердило високу точність прогнозів і спроможність у виявленні нештатних ситуацій у даних.

Побудовано програмну архітектуру системи, зокрема на базі модульного та мікросервісного підходів. Кожна підсистема спроможна функціонувати як

окремий компонент, отже, забезпечується незалежність оновлень, гнучкість масштабування разом із високим рівнем відмовостійкості.

Утворено комунікаційну підсистему, що підтримує сучасні протоколи MQTT, OPC UA та REST API. До того ж, завдяки їй забезпечується низька затримка передавання даних, а також надійна синхронізація між модулями та стабільність роботи навіть у разі втрати окремих пакетів.

ПЗ розроблено з урахуванням відкритих технологій. Таким чином, це гарантувало мінімальну вартість упровадження та портативність, а також сумісність із промисловими системами.

Окрім того, було опрацьовані питання щодо інформаційної безпеки та відмовостійкості. Так, було задіяно низку процесів: шифрування трафіку, розмежування доступу, захист від типових атак (SQLi, MITM, XSS) разом із механізмами автоматичного відновлення після збоїв.

До того ж, було упорядковано повний комплект програмної документації. Насамперед йдеться про архітектурні діаграми, алгоритмічні схеми, описи модулів, UML-діаграми та структуру БД, адже вони є супровідним підтвердженням, а також підґрунтям для подальшого розроблення системи.

Загалом третій розділ було присвячено розробленню завершеної алгоритмічної та програмної моделі автоматизованої системи. Вона, зі свого боку, повністю відповідає окресленим функціональним і нефункціональним вимогам, а також забезпечує роботу в реальному часі. Водночас підтримує високоточну аналітику й орієнтована на сучасні стандарти побудови кіберфізичних виробничих систем.

## 5 БЕЗПЕЧНІ УМОВИ ПРАЦІ У ЛАБОРАТОРНОМУ ПРИМІЩЕНІ

Розміри навчальної лабораторії, в якому проводилися дослідження з кваліфікаційної роботи складає  $8,3 \times 6 \times 3,1$  м.

Робоче місце являє собою стіл, який обладнаний персональним комп'ютером (ПК), потужність якого становлює – 600 Вт, з монітором LCD – 24”, частотою розгортки – 60 Гц.

У приміщенні працює 2 людини. Площа приміщення  $49,8 \text{ м}^2$ , об'єм –  $154,38 \text{ м}^4$ . Згідно НПАОП 0.00-1.28-10 площа на одне робоче місце має становити не менше  $6 \text{ м}^2$ , а об'єм –  $20 \text{ м}^4$ .

Згідно НПАОП 40.1-1.21-98 приміщення можна віднести до категорії без підвищеної небезпеки (сухе не запилене приміщення з ізолюючим підлогою) [37].

В приміщенні використовується система живлячих провідників, трифазна, чотирипровідна з глухо заземленою нейтраллю напругою до 1000 В, тому, згідно НПАОП 40.1-1.32-01 використовується система заземлення TN-C-S типу. Додатково застосовується повторне заземлення нульового проводу з метою зниження потенціалу корпусів і напруги дотику у випадках обриву нульового проводу [37].

Робота в приміщенні проводиться сидячи і не вимагає фізичної напруги. Тому вона відноситься до категорії Ia (легкі фізичні роботи, енерговитрати до 120 ккал/ч). З метою забезпечення комфортних умов для працівників та відповідно до ДСН 4.4.6.042-99 в приміщенні встановлені наступні оптимальні показники мікроклімату:

– для холодного періоду: температура повітря від  $22 \text{ }^\circ\text{C}$  до  $24 \text{ }^\circ\text{C}$ ; вологість повітря від 40 до 60 %; швидкість руху повітря оптимальна  $0,1 \text{ м/с}$ ;

– для теплого періоду року: температура повітря  $23 - 25 \text{ }^\circ\text{C}$ ; вологість повітря 40 – 60 %; швидкість руху повітря оптимальна  $0,1 \text{ м/с}$ , допустима  $0,2 \text{ м/с}$ .

Приміщення лабораторії має природне та штучне освітлення відповідно до ДБН Ст. 2.5-28-2006.

Природне світло проникає через бічні світлопроєми, зорієнтовані на північ, і забезпечують коефіцієнт природної освітленості (КПО) не нижче 1,2 %.

Необхідна освітленість приміщення 200 – 500 лк [37].

Згідно ДСН 4.4.6.037-99 рівень шуму в лабораторії не перевищує 50 дБА.

Категорія приміщення П – Па згідно ПУЕ-2011. Лабораторія розташована в будівлі II ступеня вогнестійкості згідно з ДБН 1.1.7-2002. По пожежо- і вибухонебезпечності відноситься до категорії В відповідно до НАПБ Б.04.002-2007 (у приміщенні лабораторії знаходяться тверді горючі матеріали).

В приміщенні знаходяться 3 робочих місця, оснащених ПК. При підвищенні температури окремих вузлів можливо оплавлення ізоляції сполучних проводів, яке веде до замикання, що супроводжується в свою чергу щирій.

Пожежна безпека в лабораторії забезпечується відповідно до ГОСТ 12.1.004-91, системою запобігання пожежі, протипожежного захисту і організаційно-технічними заходами: встановлено 1 вогнегасник ВВК-1,4 (з розрахунку один на 3 ПК); приміщення лабораторії обладнано системою автоматичної пожежної сигналізації з застосуванням димових датчиків типу ДИП-1 (1 штука) згідно з ДБН Ст. 2.5-56-2010 (1 на 86 м<sup>2</sup>); навчання персоналу правилам пожежної безпеки, регулярні інструктажі з пожежної безпеки; розміщені інструкції та план евакуації персоналу у випадку пожежі [37].

## ВИСНОВКИ

У результаті виконання досліджень за темою кваліфікаційної роботи було проведено комплексне дослідження, спрямоване на розроблення автоматизованої системи інформаційно-технічного забезпечення кіберфізичного виробництва. Отримані результати підтверджують доцільність застосування сучасних інформаційних технологій, засобів автоматизації та комунікаційних протоколів для підвищення ефективності виробничих процесів.

Проаналізовано сутність і структуру КФС, визначено їх основні компоненти – сенсорні підсистеми, контролери, мережеві інтерфейси, обчислювальні модулі та програмні сервіси. Обґрунтовано роль взаємодії фізичних процесів із цифровими моделями у забезпеченні адаптивності та гнучкості виробництва.

Розроблено архітектуру автоматизованої системи, що поєднує рівні збору, обробки, передавання та аналізу даних у єдиному інформаційному середовищі. Запропонована модель охоплює польовий, проміжний, серверний і корпоративний рівні, що забезпечує наскрізну інтеграцію виробничих процесів.

Сформовано методичні підходи до побудови інформаційно-технічного забезпечення кіберфізичного виробництва на основі принципів відкритості, масштабованості, безпеки та сумісності.

Реалізовано алгоритми збору та передачі даних з використанням промислових протоколів OPC UA та MQTT, що забезпечують взаємодію між фізичними пристроями, серверними системами та корпоративними додатками.

Розроблено механізми інтеграції з зовнішніми інформаційними системами (ERP, MES, SCADA) через REST API, що дозволяє формувати єдиний цифровий простір підприємства та підвищувати ефективність прийняття управлінських рішень.

Створено імітаційну модель кіберфізичної системи виробництва, за допомогою якої перевірено працездатність розроблених алгоритмів і підтверджено стабільність роботи системи в режимі реального часу.

Проведено оцінювання ефективності системи, яке показало зниження часу реакції системи керування, покращення якості моніторингу технологічних процесів і зменшення ризику виникнення аварійних ситуацій.

Прораховано штучне освітлення в дослідницькій лабораторії, де виконувалась кваліфікаційна робота щодо розроблення системи автоматизації інформаційно-технічного забезпечення кіберфізичного виробництва.

Наукова новизна отриманих результатів полягає у розробленні узагальненої моделі автоматизованої системи кіберфізичного виробництва, яка інтегрує фізичні, інформаційні та обчислювальні компоненти на основі гібридної архітектури зв'язку OPC UA – MQTT – REST API.

Практичне значення результатів роботи полягає в можливості впровадження розробленої системи на підприємствах промислового профілю для забезпечення цифровізації виробничих процесів, підвищення рівня автоматизації, моніторингу й контролю якості.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Методичні вказівки з підготовки та захисту кваліфікаційної роботи здобувачами другого (магістерського) рівня вищої освіти спеціальності 174 Автоматизація, комп'ютерно-інтегровані технології та робототехніка, освітньо-професійних програм: «Комп'ютерно-інтегровані технологічні процеси і виробництва», «Комп'ютеризовані та робототехнічні системи» /Упоряд. І. Ш. Невлюдов, Р. В. Артюх, В. В. Безкоровайний, Н. П. Демська, В.В.Євсєєв, О. І. Филипенко, О. М. Цимбал. Харків: ХНУРЕ, 2024. – 49 с.

2. Положення про кваліфікаційну роботу здобувача вищої освіти на другому (магістерському) рівні [Електронний ресурс]: Наказ ХНУРЕ від 06 травня 2021 р. No 144. – Режим доступу: [https://nure.ua/wpcontent/uploads/Main\\_Docs\\_NURE/143-vid-06.05.2021-pro-vvedennja-v-dijurishennja-vchenoi-radi-universitetu.pdf](https://nure.ua/wpcontent/uploads/Main_Docs_NURE/143-vid-06.05.2021-pro-vvedennja-v-dijurishennja-vchenoi-radi-universitetu.pdf).

3. ДСТУ 3008-15. Документація. Звіти у сфері науки та техніки. структура та правила оформлення. Введ. 2015-06-22. К. Держстандарт України, 2017. – 29 с.

4. Hubskiy Ihor, Osman Amira, Allakhveranov Rauf. Manipulators with elastic joints and their control systems // I International scientific and practical conference “Global trends in the development of information technology and science” Stockholm, Sweden. International Scientific Unity. 2025. pp. 217-220, ISBN 979-8-89704-992-9. Режим доступу: <https://doi.org/10.70286/isu-08.01.2025>.

5. Євсєєв В. В. Моделі та методи автоматизації кіберфізичних виробничих систем // Вісник ХНУРЕ. 2021. № 3(99). С. 45-52.

6. Нікітін О. В. Інформаційно-технічне забезпечення виробничих систем у контексті Індустрії 4.0 // Системи управління, навігації та зв'язку. 2020. Т. 2(60). С. 66-71.

7. Колосюк О. П., Рябенко В. О. Кіберфізичні системи та технології індустрії 4.0 // Наукові праці Одеської нац. академії зв'язку. 2019. Вип. 1. С. 57-64.
8. Соколенко В. Ф., Дорошенко О. Г. Інтеграція КФС та ERP-систем на основі моделі RAMI 4.0 // Вісник НТУ «ХПІ». 2022. Серія: Інформатика та моделювання. С. 25-31.
9. Lee E. A. Cyber-physical systems – Are computing foundations adequate? // NSF Workshop on Cyber-Physical Systems. 2006. P. 1-9.
10. Lee J., Bagheri B., Kao H. A. A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems // Manufacturing Letters. 2015. Vol. 4. P. 18-24.
11. Monostori L. Cyber-Physical Production Systems: Roots, Expectations and R&D Challenges // Procedia CIRP. 2014. Vol. 17. P. 9-14.
12. Rajkumar R., Lee I., Sha L., Stankovic J. Cyber-Physical Systems: The Next Computing Revolution // Proc. DAC. 2010. P. 731-736.
14. Karnouskos S., Colombo A. W. Industrial automation based on Cyber-Physical Systems technologies // IFAC PapersOnLine. 2015. Vol. 48(3). P. 3000-3005.
14. Oks S. J., Jalowski M., Lechner M. Cyber-Physical Systems in the Context of Industry 4.0: A Review // Information Systems Frontiers. 2022. Vol. 26. P. 1731-1772.
15. Suvarna M. et al. Cyber-Physical Production Systems for Data-Driven Manufacturing: A Review // Procedia CIRP. 2021. Vol. 100. P. 495-501.
16. Müller T. et al. Cyber-Physical Production Systems: Enhancement with a Reconfiguration Capability // Computers in Industry. 2021. Vol. 129. P. 103-118.
17. Льюнг Л. Ідентифікація систем: теорія для користувача. – К.: Наукова думка, 2020. – 512 с.
18. Огарков С. Л. Математичні моделі в системах автоматизації: методи побудови і дослідження. – Львів: ЛНУ, 2021. – 298 с.

19. Катасонов Н. А. Математичне моделювання складних систем: аналітичні методи. – Харків: ХНАМГ, 2020. – 336 с.
20. Зайченко Ю. П. Дослідження операцій: моделі та методи. – Київ: КНЕУ, 2020. – 312 с.
21. Ковалишин В. В. Методи аналізу та моделювання динамічних систем. – Львів: Видавництво ЛНТУ, 2018. – 264 с.
22. Катсанюк С. М., Лисенко В. А. Моделювання та алгоритми цифрових систем керування. – Київ: КНУТД, 2019. – 284 с.
23. Крейг Дж. Вступ до робототехніки: механіка та алгоритми керування. – К.: Видавництво КП, 2021. – 448 с.
24. Seborg D., Edgar T., Mellichamp D. Process Dynamics and Control. 4th ed. – Wiley, 2016. – 492 p.
25. Zhang L., Wang S. Cyber-Physical Systems: Architecture, Modeling and Control. – Springer, 2020. – 380 p.
26. Кузнецов М. Ю. Інтелектуальні системи керування: алгоритми та програмні засоби. – Харків: ХНУРЕ, 2022. – 312 с.
27. OPC Unified Architecture. Part 1–14. OPC Foundation Specifications. – 2019.
28. MQTT Version 4.1.1. OASIS Standard. – 2014. – Режим доступу: <https://mqtt.org>.
29. Docker Documentation. Docker Inc., 2024. – Режим доступу: <https://docs.docker.com/>.
30. Python Software Foundation. Python 3 Documentation. – 2024. – Режим доступу: <https://docs.python.org/>.
31. IEC 61131-4. Programmable Controllers – Programming Languages. – International Electrotechnical Commission, 2017.
32. Schneider Electric. Modicon ПЛК: Programming Guide. – 2021.
33. IEEE Standard for Cyber-Physical Systems (КФС). IEEE 21834-2022.
34. Пархоменко О. Ф. Методи оброблення даних у системах автоматизації. – Львів: ЛНУ, 2020. – 276 с.

35. Куваєв О. М., Литвиненко І. В. Математичне моделювання технічних систем. – Київ: Видавництво КПІ, 2020. – 356 с.

36. Сорокоумова Н. В., Сухоруков М. Г. Теорія автоматичного керування: Математичні моделі та методи аналізу. – Харків: ХНУРЕ, 2019. – 412 с.

37. Комплекс навчально-методичного забезпечення навчальної дисципліни «Організація керування умовами праці» підготовки освітнього рівня бакалавр усіх спеціальностей та усіх напрямів університету [Електронний ресурс] / ХНУРЕ; розроб.: Т. Є. Стиценко, Г. В. Пронюк, Н. М. Сердюк. – Харків, 2017. – 108 с.