

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр  
Освітній рівень

Кіберфізична система промислових послуг у виробництві  
Назва теми

КвРКІ.210297.20.02.10 ПЗ  
Шифр

Галузь знань 12 «Інформаційні технології»

Шифр, назва

Спеціальність 123 «Комп'ютерна інженерія»

Шифр, назва

Освітня програма «Комп'ютерна інженерія та програмування»

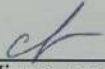
Назва

Виконав: студент IV курсу, група KI2-20-2

  
Підпис

А. М. Крушельницький  
Ініціали, прізвище

Керівник

  
Підпис, дата

С. І. Саченко  
Ініціали, прізвище

Нормоконтролер

  
Підпис, дата

С. М. Лисенко  
Ініціали, прізвище

До захисту допускаю:  
Зав. кафедри комп'ютерної  
інженерії та інформаційних  
систем

  
Підпис

Т. О. Говорущенко  
Ініціали, прізвище

«24» червня 2024 р.

Хмельницький 2024

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ  
Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ  
Освітній рівень БАКАЛАВР  
Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ  
Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ  
Освітня програма «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ ТА ПРОГРАМУВАННЯ»

ЗАТВЕРДЖУЮ

Зав. кафедри Т.О.Говорущенко

“ 10 ” 01 2024 р.

**ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА**

Крушельницькому Андрію Михайловичу

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Кіберфізична система промислових послуг у виробництві

Керівник проекту (роботи) Саченко С. І., к.е.н, доцент.

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 15.02.2024 р. № 8

2. Строк подання студентом проекту (роботи) на кафедру 01.06.2024 р.

3. Вихідні дані до проекту (роботи) Завдання на кваліфікаційну роботу

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) \_\_\_\_\_

Аналіз сучасних рішень для вирішення задач

Обґрунтування вибору компонентів та середовища реалізації

Програмна реалізація та результати дослідження

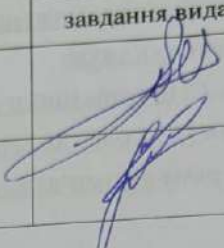
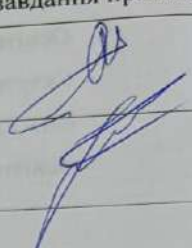
5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень) \_\_\_\_\_

Схеми системи

Схема виконання програм як інформаційних потоків

Візуалізація температурного профілю

6. Консультанти розділів дипломного проекту (роботи)

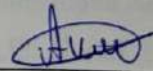
Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Лисенко С.М., професор кафедри КПС		
Антиплагіат	Нічепорук А.О., доцент кафедри КПС		

7. Дата видачі завдання « 10 » 01 2024 р.

**КАЛЕНДАРНИЙ ПЛАН**

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напряму дослідження та узгодження тематики кваліфікаційної роботи з керівником	10.01.2024	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.02.2024	виконано
3	Робота над розділом 1 – аналіз сучасних рішень для вирішення задач	01.03.2024	виконано
4	Робота над розділом 2 – обґрунтування вибору компонентів та середовища реалізації	01.04.2024	виконано
5	Робота над розділом 3 – програмна реалізація та результати дослідження	29.04.2024	виконано
6	Оформлення пояснювальної записки згідно вимог	25.05.2024	виконано
7	Попередній захист ВКР	30.05.2024	виконано
8	Захист ВКР на засіданні ЕК	Червень 2024 року	

Студент



Підпис

А.М. Крушельницький

Ініціали, прізвище

Керівник роботи



Підпис

С. І. Саченко

Ініціали, прізвище

## АНОТАЦІЯ

Тема кваліфікаційної роботи: «Кіберфізична система промислових послуг у виробництві».

Автор роботи: Крушельницький А. М.

Керівник роботи: Саченко С. І.

Пояснювальна записка: 57 с., 10 рис., 5 табл., 2 дод., 60 джерел.

Графічна частина: 3 креслення.

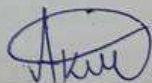
### ІНФОРМАЦІЙНА СИСТЕМА, КІБЕРФІЗИЧНА СИСТЕМА, ПРОМИСЛОВІ ПОСЛУГИ

Мета кваліфікаційної роботи є дослідження та аналіз сучасних рішень для інтеграції кіберфізичних систем (КФС) у промислових послугах виробництва, а також обґрунтування вибору оптимальних компонентів і середовища реалізації таких систем.

Об'єктом дослідження є кіберфізичні системи, інтегровані у промислові послуги виробництва. Це включає в себе технології та платформи, що забезпечують взаємодію фізичних процесів з цифровими системами, такі як Інтернет речей (ІоТ), великі дані, штучний інтелект (АІ) та хмарні обчислення.

Завдання роботи: провести аналіз поточного стану та сучасних рішень для інтеграції кіберфізичних систем у промислових послугах виробництва, обґрунтувати вибір компонентів та середовища реалізації для розробки кіберфізичних систем, розробити програмну реалізацію КФС на основі обраних компонентів, оцінити ефективність впровадженого рішення через практичні дослідження та аналіз отриманих результатів.

Для досягнення поставленої мети використовуються такі методи як методи дослідження, аналізу літератури, моделювання та симуляції, програмування та розробки, експериментів, системного аналізу та проектування.







Підпис студента

30.05.2024

Дата

## ЗМІСТ

<b>ВСТУП</b> .....	<b>3</b>
<b>1 АНАЛІЗ СУЧАСНИХ РІШЕНЬ ДЛЯ ВИРІШЕННЯ ЗАДАЧ</b> .....	<b>5</b>
1.1 Поточний стан кіберфізичних систем у промислових послугах виробництва.....	5
1.2 Аналіз існуючих рішень та підходів до реалізації кіберфізичних систем у виробництві.....	10
1.3 Висновки до розділу.....	18
<b>2 ОБҐРУНТУВАННЯ ВИБОРУ КОМПОНЕНТІВ ТА СЕРЕДОВИЩА РЕАЛІЗАЦІЇ</b> .....	<b>22</b>
2.1 Вимоги до компонентів кіберфізичних систем.....	22
2.2 Вибір середовища реалізації.....	34
2.3 Забезпечення безпеки та масштабованості кіберфізичних систем.....	41
2.4 Висновок до розділу.....	45
<b>3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ</b> .....	<b>46</b>
3.1 Розробка алгоритмів для збору та аналізу даних.....	46
3.2 Розробка веб-інтерфейсу додатку.....	49
3.3 Розробка програмного забезпечення для інтеграції КФС.....	52
3.4 Тестування та налагодження.....	54
3.5 Впровадження КФС у виробничий процес.....	58
3.6 Оцінка ефективності впровадженого рішення.....	59
<b>ВИСНОВКИ</b> .....	<b>60</b>
<b>ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ</b> .....	<b>61</b>
<b>Додаток А «Схеми системи»</b> .....	<b>65</b>
<b>Додаток Б «Схеми виконання програм як інформаційних потоків»</b> .....	<b>66</b>
<b>Додаток В «Візуалізація температурного профілю»</b> .....	<b>67</b>
<b>Додаток Г «Програмний код продукту»</b> .....	<b>68</b>
<b>Додаток Д «Програмний код навчання моделі TensorFlow»</b> .....	<b>70</b>
<b>Додаток Е «Програмний код веб-інтерфейсу»</b> .....	<b>72</b>

				КвРКІ.210297.21.02.10 ПЗ			
Зм. Арк.	№ док.ум.	Підпис	Дата	Кіберфізична система послуг у виробництві Пояснювальна записка	Літера	Аркуш	Аркушів
Виконав	Крушельницький				у	2	57
Перевір.	Саченко С.І.				ХНУ КІ2-20-2		
Н.контр.	Лисенко С.М.						
Затвер.	Говорущенко Т.О.		24.06				

## ВСТУП

Кіберфізична система (КФС) промислових послуг у виробництві - це інтегрована система, яка поєднує фізичні виробничі процеси з цифровими технологіями, забезпечуючи автоматизоване управління та оптимізацію виробничих операцій. Основними складовими КФС є датчики, виконавчі механізми, мережеві комунікації, програмне забезпечення та аналітичні інструменти.

У сучасному світі швидкий розвиток технологій призводить до значних змін у різних галузях промисловості. Кіберфізичні системи (далі - КФС) стають одним із ключових елементів четвертої промислової революції, забезпечуючи інтеграцію фізичних і цифрових процесів. Впровадження КФС у виробничі процеси дозволяє підвищити ефективність, гнучкість та автоматизацію, що є надзвичайно важливим для збереження конкурентоспроможності підприємств на світовому ринку. Саме тому дослідження можливостей та ефективності застосування КФС у промислових послугах виробництва є надзвичайно актуальним.

Проект має можливість розвитку, модернізації, тому розробка і створення пристрою, представлених в даній дипломній роботі є доцільним рішенням як з технічного так і з економічного боку.

Метою даної роботи є дослідження та впровадження кіберфізичних систем у промислові послуги виробництва для підвищення ефективності, автоматизації та гнучкості виробничих процесів. Для досягнення поставленої мети було визначено такі завдання:

- 1) провести детальний аналіз існуючих рішень для впровадження КФС у виробничі процеси;
- 2) обґрунтувати вибір компонентів та середовища для реалізації КФС, враховуючи аспекти безпеки, масштабованості та сумісності;
- 3) розробити та впровадити програмне забезпечення для інтеграції КФС у виробничий процес;

					КВРКІ.21/0297.21.02.10 ПЗ	Арк.
						3
Зм.	Арк.	№ докум.	Підпис	Дата		

4) оцінити ефективність впровадженого рішення через практичні дослідження та аналіз отриманих результатів.

У результаті успішної реалізації дослідження передбачається отримання значних покращень у виробничих процесах, зокрема, підвищення продуктивності, зниження витрат та покращення якості продукції.

Дослідження також передбачає аналіз можливих ризиків та впровадження заходів з їх запобігання.

У подальшому розвитку дослідження можна розглядати можливість розширення області застосування кіберфізичних систем на інші галузі промисловості та розробку нових функціональних можливостей, таких як автоматизація моніторингу та діагностики стану обладнання.

					КВРКІ.21/0297.21.02.10 ПЗ	Арк.
						4
Зм.	Арк.	№ докум.	Підпис	Дата		

# 1 АНАЛІЗ СУЧАСНИХ РІШЕНЬ ДЛЯ ВИРІШЕННЯ ЗАДАЧ

## 1.1 Поточний стан кіберфізичних систем у промислових послугах виробництва

Поточний стан кіберфізичних систем (КФС) у промислових послугах виробництва відображає загальну динаміку та тенденції їх впровадження та розвитку [1]. Ключові моменти загального стану:

- 1) зростання зацікавленості та впровадження;
- 2) інтеграція з іншими технологічними тенденціями;
- 3) посилення конкуренції та вимоги до інновацій;
- 4) виклики та перешкоди;
- 5) потреба у гнучкості та адаптивності;
- 6) розвиток технологій;
- 7) потенціал зменшення витрат;
- 8) конкуренція;
- 9) зростання обсягу даних.

Загальною тенденцією впровадження КФС у промислових послугах виробництва є поступове переходження [2] до більш інтегрованих [3], ефективних та гнучких систем [4], які допомагають підприємствам забезпечувати конкурентні переваги у сучасному ринковому середовищі. Перехід до більш інтегрованих систем означає, що КФС стають важливою складовою виробничих процесів, які покликані сприяти оптимальній координації та співпраці між різними компонентами системи виробництва. Головною метою переходу до ефективних КФС є покращення продуктивності та оптимізація виробничих процесів. Це може включати автоматизацію рутинних завдань, покращення контролю за процесами та оптимізацію використання ресурсів [1]. Сучасні підприємства розуміють важливість гнучкості та адаптивності у змінному ринковому середовищі. Більш гнучкі та адаптивні КФС дозволяють підприємствам швидко реагувати на зміни [5] в запитах ринку та внутрішніх умовах. Впровадження інноваційних технологій, таких як КФС, дозволяє підприємствам створювати конкурентні переваги на ринку.

					КВРКІ.21/0297.21.02.10 ПЗ	Арк.
						5
Зм.	Арк.	№ докум.	Підпис	Дата		

Ці системи допомагають вдосконалювати продукцію, зменшувати час виробництва та виготовляти якісні продукти [6]. Завдяки інтегрованим та ефективним системам КФС [7], підприємства можуть досягати вищих стандартів якості продукції та безпеки виробничих процесів [8]. Це робить їх більш конкурентоспроможними на ринку та сприяє збільшенню задоволеності клієнтів (рис. 1.1).



Рисунок 1.1 - Зацікавленість підприємств у впровадженні кіберфізичних систем

Останні роки свідчать про значний ріст зацікавленості підприємств у впровадженні КФС [9]. Промислові компанії [10] у всьому світі активно впроваджують ці системи для оптимізації виробничих процесів та підвищення ефективності. З впровадженням ІоТ та інших сенсорних технологій [11], компанії збирають значні обсяги даних [12] про свої виробничі процеси [13]. Це створює потребу в аналізі [14] та використанні цих даних для оптимізації [15] ефективності та виробничих процесів [16]. У світі конкуренція між підприємствами стає все більш жорсткою. Впровадження КФС може стати ключовим фактором для

підприємств у збереженні конкурентоспроможності та виграванні на ринку. КФС дозволяють автоматизувати багато виробничих процесів, що може призвести до зменшення витрат на працю та енергію. Це може бути особливо важливим у сферах з високою конкуренцією та низькими маржами прибутку. Завдяки постійному розвитку технологій у сфері IoT, AI та аналітики даних, впровадження КФС стає більш доступним. Сучасні ринкові умови часто змінюються, і підприємствам потрібно бути гнучкими та адаптивними. Впровадження КФС [17] може допомогти підприємствам швидко реагувати на зміни та впроваджувати нові стратегії та технології підприємств. Це стимулює їхнє зацікавлення у використанні цих систем. У цілому, зростання зацікавленості підприємств у впровадженні КФС свідчить про їхнє бачення цих систем як ключового елемента для досягнення конкурентної переваги та підвищення ефективності виробництва.

Реальні приклади успішного впровадження КФС включають автоматизацію виробничих ліній, моніторинг стану обладнання за допомогою датчиків IoT, використання аналітики великих даних для прогнозування збоїв та оптимізації процесів, а також використання штучного інтелекту для підвищення точності та швидкості виробничих операцій. КФС можуть бути використані для автоматизації виробничих ліній [18], що дозволяє підприємствам значно знизити ручну працю, підвищити продуктивність та знизити ймовірність помилок. Наприклад, роботизовані системи можуть автоматично здійснювати монтаж, упаковку або сортування товарів на виробничій лінії. Використання датчиків Internet of Things (IoT) дозволяє в реальному часі відстежувати стан обладнання та виробничих процесів. Це дозволяє оперативно виявляти проблеми, запобігати аваріям та збоїв, а також планувати технічне обслуговування для підтримки безперебійної роботи обладнання. Аналітика великих даних [19] дозволяє збирати, аналізувати та використовувати великі обсяги даних для прогнозування збоїв у виробничих процесах та оптимізації робочих потоків. Наприклад, за допомогою аналізу даних можна виявляти патерни в роботі обладнання, що вказують на його можливі збої або потребу у технічному обслуговуванні [20]. Штучний інтелект може бути використаний для автоматизації та оптимізації різних виробничих процесів.

					КВРКІ.21/0297.21.02.10 ПЗ	Арк. 7
Зм.	Арк.	№ докум.	Підпис	Дата		

Наприклад, системи машинного навчання можуть навчитися розпізнавати аномалії у виробничих даних, покращуючи якість контролю якості та допомагаючи уникнути виробничих збоїв.

Постійний розвиток технологій IoT, великих даних, штучного інтелекту та хмарних обчислень стимулює появу нових можливостей для впровадження КФС. Інновації в галузі сенсорів, збірників даних, аналітичних платформ та систем управління дозволяють створювати більш ефективні та адаптивні КФС, які можуть адаптуватися до змінних умов виробництва [21].

Розвиток нових технологій дозволяє створювати більш потужні та мініатюрні сенсори, які можуть збирати дані про стан обладнання та виробничих процесів у реальному часі. Ці сенсори забезпечують точність і швидкість збору інформації, що є критично важливим для підтримки ефективного та безпечного функціонування виробничих систем.

Використання технологій обробки даних на краю дозволяє проводити аналіз та обробку даних безпосередньо на місці їхнього виникнення. Це значно зменшує завантаження на централізовані обчислювальні ресурси, оскільки дані обробляються локально, без необхідності їх передачі до віддалених серверів. Обробка на краю також зменшує затримки в реакції систем на зміни у виробничому процесі.

Постійний розвиток аналітичних платформ дозволяє виконувати складний аналіз великих обсягів даних, виявляти патерни та тренди, що допомагає в прийнятті стратегічних рішень у виробничому середовищі.

Поява нових алгоритмів машинного навчання та штучного інтелекту дозволяє використовувати великі дані для прогнозування збоїв у виробництві [22] та оптимізації робочих потоків. Застосування алгоритмів машинного навчання та нейронних мереж дозволяє прогнозувати виробничі збої, оптимізувати робочі процеси та підвищувати ефективність [23] обладнання.

У таблиці 1.1 показано узагальнену інформацію про те, які конкретні переваги приносять інновації в області Інтернету речей (IoT), великих даних,

					КВРКІ.21/0297.21.02.10 ПЗ	Арк.
						8
Зм.	Арк.	№ докум.	Підпис	Дата		

штучного інтелекту та хмарних обчислень для впровадження кіберфізичних систем у промислових послугах виробництва.

Таблиця 1.1 – Узагальнена інформація про інновації

№	Технологія (Інновація)	Переваги для КФС
1	Інтернет речей (IoT)	Збір реального часу даних, автоматизація моніторингу обладнання
2	Великі дані (Big Data)	Аналіз та прогнозування збоїв, оптимізація виробничих процесів
3	Штучний інтелект (AI)	Підвищення точності та швидкості виробничих операцій, прогнозування збоїв
4	Хмарні обчислення	Гнучкість та масштабованість обчислювальних ресурсів, забезпечення доступу до даних

Незважаючи на широкий потенціал, пов'язаний з використанням кіберфізичних систем (КФС) у промисловості, існують ряд викликів, які потребують уваги. Один з них - це питання безпеки даних, яке стає особливо актуальним у зв'язку з розширенням збору та обробки великих обсягів інформації.

Іншим важливим аспектом є інтеграція КФС з вже наявними системами у промисловому середовищі. Це може вимагати значних зусиль та витрат на забезпечення сумісності та оптимальної взаємодії між різними технологічними рішеннями.

Нестабільність деяких технологій, їх швидка зміна та постійне оновлення також ускладнюють впровадження КФС у промисловість. Це може вимагати постійного моніторингу та адаптації до нових ринкових умов та технологічних змін.

Одним з ключових викликів є високі витрати на впровадження КФС. Це може стати перешкодою для деяких підприємств, особливо для малих та середніх підприємств, які можуть бути обмежені бюджетними обмеженнями.

Незважаючи на ці виклики, застосування технологій Internet of Things (IoT), великих даних, штучного інтелекту та хмарних обчислень у промисловому виробництві вже показує значні результати у вигляді оптимізації та автоматизації виробничих процесів.

З розвитком технологій та набуттям досвіду їх впровадження, КФС стають все більш важливим елементом конкурентоспроможності для підприємств у промисловому секторі, але для цього необхідно активно працювати над вирішенням вищезгаданих викликів та забезпечити ефективне управління технологічними змінами.

## 1.2 Аналіз існуючих рішень та підходів до реалізації кіберфізичних систем у виробництві

Основна мета аналізу полягає в ідентифікації найбільш ефективних та оптимальних рішень для впровадження КФС у виробничому середовищі [26]. Для цього проводиться детальний огляд існуючих підходів, методик та технологій, які можуть бути використані для реалізації кіберфізичних систем. Дослідження різних технологічних платформ, які використовуються для побудови кіберфізичних систем, таких як платформи IoT, хмарні обчислення, аналітичні системи тощо. Оцінка методів аналізу великих обсягів даних, що збираються від датчиків та інших джерел у виробничих процесах. Це може включати методи машинного навчання, аналізу часових рядів, статистичний аналіз тощо. Аналіз існуючих рішень та підходів до реалізації кіберфізичних систем у виробництві допомагає підприємствам обрати оптимальний шлях для впровадження та використання цих систем з метою підвищення ефективності виробничих процесів та конкурентоспроможності [27]. Для реалізації детального аналізу існуючих рішень та підходів для впровадження кіберфізичних систем у виробництві, спочатку

					КВРКІ.21/0297.21.02.10 ПЗ	Арк. 10
Зм.	Арк.	№ докум.	Підпис	Дата		

необхідно провести огляд популярних платформ і архітектур КФС, таких як платформи для промислового Інтернету речей (IIoT) і розподілені обчислювальні системи [28].

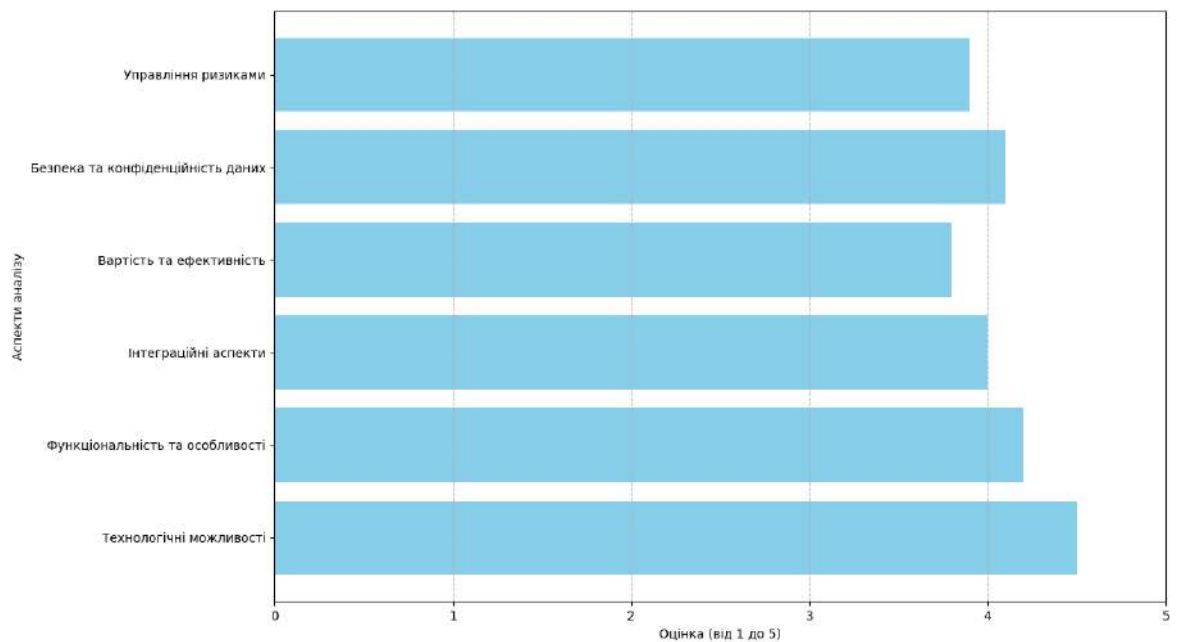


Рисунок 1.2 - Оцінка аспектів аналізу існуючих рішень у впровадженні КФС

Індустріальний Інтернет речей (IIoT) стає все більш важливим компонентом сучасних виробничих процесів [28], дозволяючи підприємствам збирати, аналізувати та використовувати дані з пристроїв та обладнання для оптимізації ефективності виробництва та прийняття більш обґрунтованих рішень. Провідні платформи для IIoT та їх можливості:

1) AWS IoT Amazon Web Services пропонує різноманітні сервіси для IIoT, включаючи пристрої AWS IoT, AWS IoT Greengrass і AWS IoT Core. Ці сервіси дозволяють забезпечити з'єднання, безпеку та управління пристроями в масштабах великих виробничих середовищ [29];

2) платформа Microsoft Azure IoT надає набір інструментів для збору, аналізу та використання даних [30] з IIoT-пристроїв. Azure IoT Hub дозволяє керувати підключеними пристроями, а Azure IoT Edge дозволяє розгортати обробку даних на місці;

3) Google Cloud IoT надає інструменти для збору та обробки даних [31] з підключених пристроїв, таких як Google Cloud IoT Core. Google також пропонує інтеграцію з іншими своїми сервісами, такими як BigQuery та TensorFlow, для аналізу та використання даних.

IBM Watson IoT пропонує рішення для IIoT, включаючи Watson IoT Platform, яка дозволяє збирати, аналізувати та використовувати дані з підключених пристроїв [32]. Крім того, IBM пропонує інтеграцію зі штучним інтелектом та аналітичними інструментами для виняткового аналізу даних.

Кожна з цих платформ має свої унікальні особливості та переваги, але всі вони дозволяють побудувати розподілені системи моніторингу та управління виробництвом.

Ці платформи надають інструменти для збору та обробки даних [33], забезпечують безпеку з'єднань та інтеграцію з іншими системами. Проте, для успішної інтеграції з уже існуючими системами на підприємстві, необхідно провести детальний аналіз сумісності та можливостей інтеграції кожної конкретної платформи з існуючою інфраструктурою.

Важливо також враховувати витрати на впровадження та підтримку. Огляд розподілених обчислювальних систем [34] в контексті впровадження кіберфізичних систем [35] у виробництві є важливим етапом при виборі оптимального підходу до побудови системи моніторингу та управління.

Таким чином, під час вибору платформи для побудови розподіленої системи моніторингу та управління виробництвом, слід враховувати унікальні особливості та переваги кожної платформи, інструменти для збору та обробки даних, безпеку з'єднань та інтеграцію з іншими системами.

Крім того, перед впровадженням платформи необхідно провести детальний аналіз її сумісності з існуючою інфраструктурою підприємства, щоб уникнути можливих проблем. Важливо також враховувати витрати не лише на початкове впровадження платформи, але й на її довгострокову підтримку та обслуговування.

Вивчення доступних варіантів розподілених обчислювальних систем та їх можливостей для впровадження кіберфізичних систем у виробництві дозволить

					КВРКІ.21/0297.21.02.10 ПЗ	Арк. 12
Зм.	Арк.	№ докум.	Підпис	Дата		

вибрати оптимальну платформу для побудови ефективної та надійної системи моніторингу та управління виробництвом.

Аналіз технологічних рішень для реалізації кіберфізичних систем (КФС) у виробництві [36] є ключовим етапом у процесі вибору оптимального технологічного стеку. Опис розподілених обчислювальних систем подано в таблиці 1.2.

Таблиця 1.2 - Розподілені обчислювальні системи

	Проблема	Опис
1	Дослідження різних архітектур розподілених обчислювальних систем	Розподілені обчислювальні системи включають різні архітектурні підходи, такі як кластерні системи, архітектури з великою кількістю вузлів, мережі з багат шаровою топологією тощо. Дослідження різних архітектур дозволяє визначити найбільш підходящий варіант для конкретного виробничого середовища з урахуванням його потреб та обмежень.
2	Аналіз можливостей цих систем для побудови великих масштабних кіберфізичних систем	Розподілені обчислювальні системи надають можливості для побудови великих масштабних кіберфізичних систем у виробництві. Вони дозволяють розподілено обробляти великі обсяги даних з датчиків та інших джерел, виконувати складні обчислення та аналізувати ці дані в реальному часі.
3	Визначення переваг та недоліків різних архітектур	Кожна архітектура розподіленої обчислювальної системи має свої переваги та недоліки. Наприклад, кластерні системи можуть бути більш простими у встановленні та налагодженні, але вони можуть мати обмежену масштабованість. У той час як системи з великою кількістю вузлів можуть бути дуже масштабованими, але вони можуть

		вимагати складнішого управління та налаштування. Визначення переваг та недоліків кожної архітектури дозволяє обрати найбільш оптимальний варіант для конкретного виробничого середовища.
--	--	--

Під час аналізу різних технологічних рішень для реалізації КФС виробничі підприємства [37] порівнюють такі рішення, як використання великих даних, штучного інтелекту, блокчейн-технологій тощо. Порівняльний аналіз дозволяє визначити, які з цих технологій найбільш ефективні та відповідають потребам конкретного підприємства. Під час аналізу технологічних рішень важливо визначити їх переваги та недоліки. Наприклад, використання великих даних [38] може дозволити отримати цінні інсайти для оптимізації виробничих процесів, але може виникнути проблеми з безпекою та конфіденційністю даних. З іншого боку, використання штучного інтелекту може підвищити автоматизацію та точність процесів, але вимагатиме значних витрат на впровадження та навчання моделей. Крім того, аналізуватимуться аспекти безпеки, масштабованості та сумісності з існуючими системами на підприємстві. Наприклад, технологічне рішення повинно бути відповідним вимогам з безпеки даних, а також бути легко масштабованим для вирішення зростаючих потреб підприємства. В цілому, аналіз технологічних рішень для реалізації КФС у виробництві [39] допомагає підприємствам обрати оптимальний технологічний стек, що відповідає їхнім потребам та обмеженням.

Під час аналізу успішних кейсів впровадження КФС у виробництві вивчаються конкретні приклади підприємств, які здійснили успішну імплементацію цих систем. Важливо визначити, які конкретні проблеми вирішувались за допомогою КФС, які технологічні рішення [40] були застосовані та які результати були досягнуті. Під час аналізу ключових факторів успіху досліджуються різні аспекти впровадження КФС, такі як стратегічне планування, вибір технологій, організаційні зміни, комунікації зі зацікавленими сторонами тощо. Важливо з'ясувати, які саме фактори виявилися критичними для успіху кожного конкретного випадку. На основі аналізу успішних кейсів формулюються

					КВРКІ.21/0297.21.02.10 ПЗ	Арк. 14
Зм.	Арк.	№ докум.	Підпис	Дата		

перспективні напрямки для подальших досліджень та впроваджень у галузі КФС у виробництві [41]. Це може включати розвиток нових технологій, вдосконалення існуючих систем, впровадження інноваційних підходів у різних сферах виробництва тощо. Загалом, аналіз успішних кейсів та напрямків розвитку є ключовим етапом у формуванні стратегії впровадження КФС у виробництві [42], що допомагає підприємствам максимізувати вигоди від цих систем та досягати конкурентних переваг.

Частка вибору напрямків для досліджень подана на рисунку 1.3. Залежність між безпекою і масштабованістю подано на рисунку 1.4. Динаміка впровадження IoT платформи показно на рисунку 1.5. Переваги та недоліки технологічних рішень подано на рисунку 1.6.

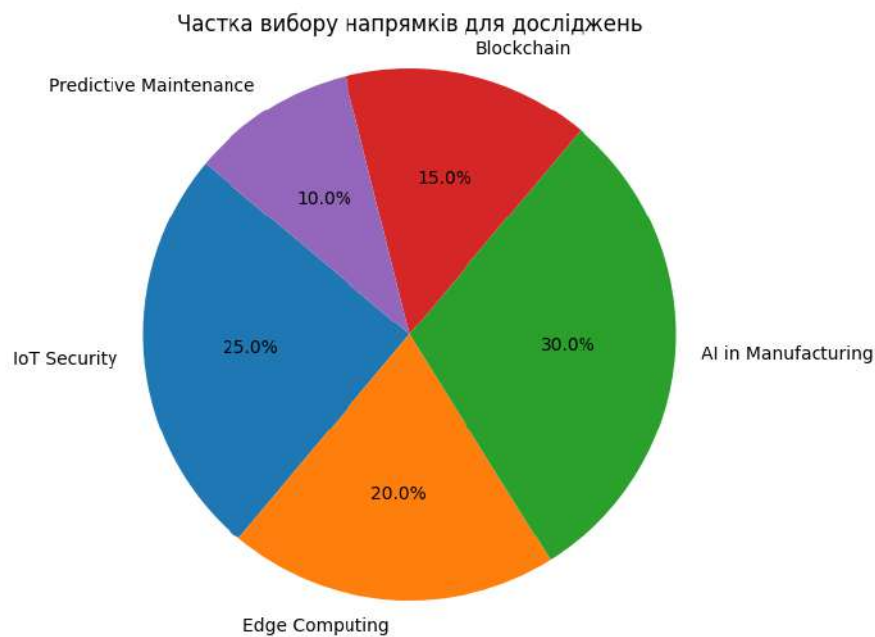


Рисунок 1.3 - Частка вибору напрямків для досліджень

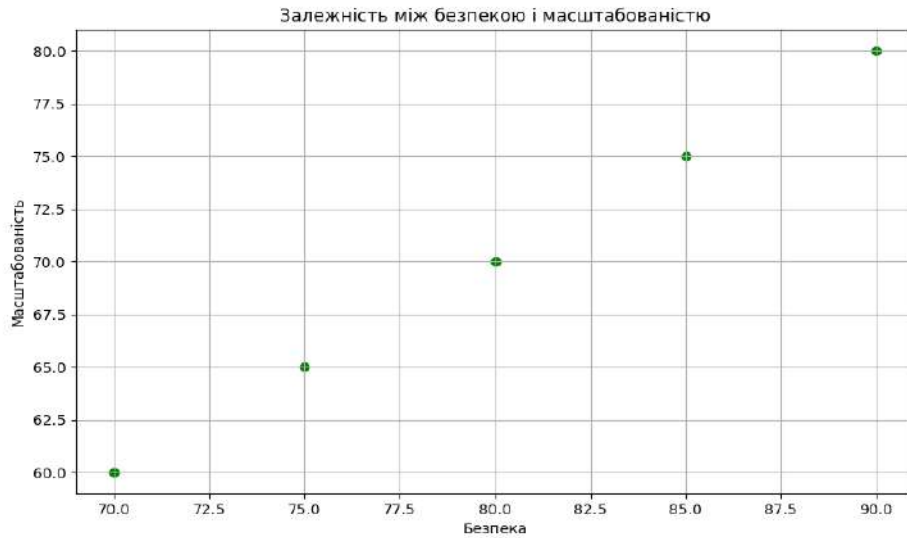


Рисунок 1.4 - Залежність між безпекою і масштабованістю

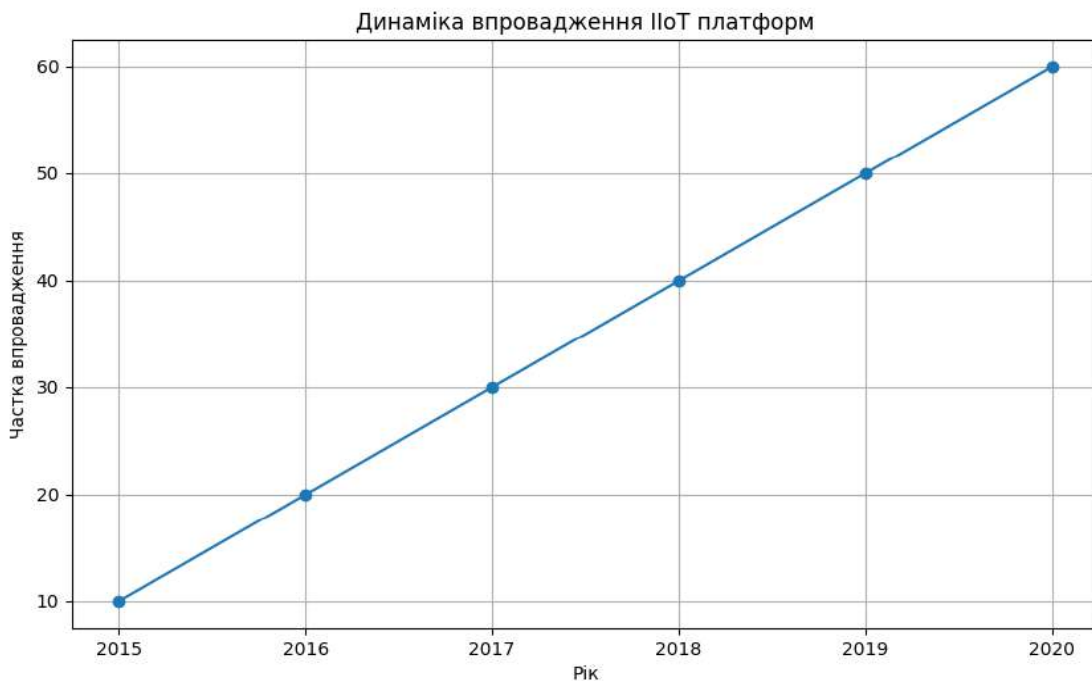


Рисунок 1.5 - Динаміка впровадження IoT платформ

Частка ринку IoT платформ подано на рисунку 1.7. Кореляція між ефективністю та витратами на впровадження КФС подано на рисунку 1.8. Динаміка впровадження КФС у виробництві подано на рисунку 1.9.

Зм.	Арк.	№ докум.	Підпис	Дата

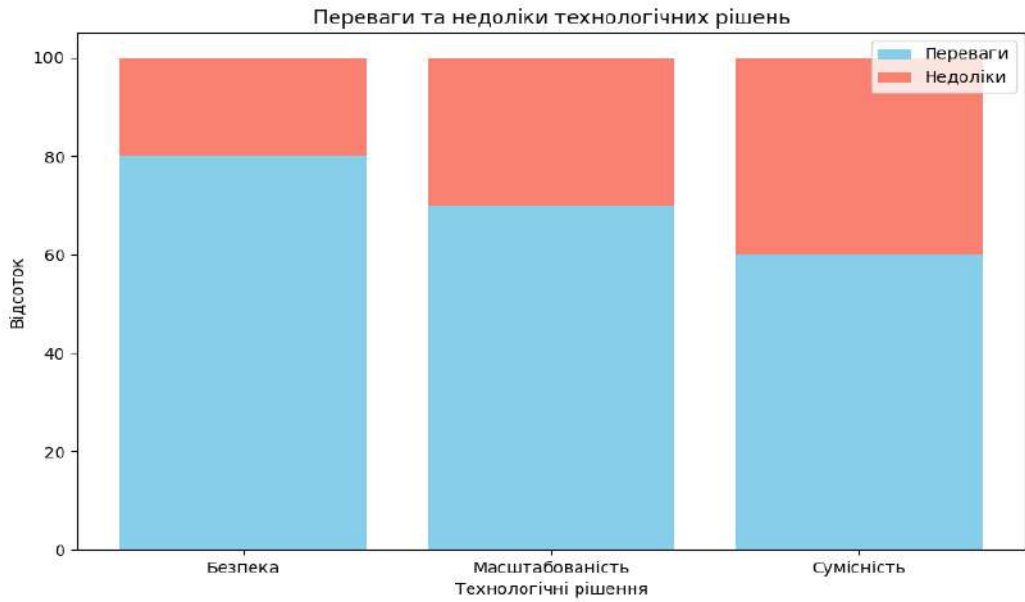


Рисунок 1.6 - Переваги та недоліки технологічних рішень

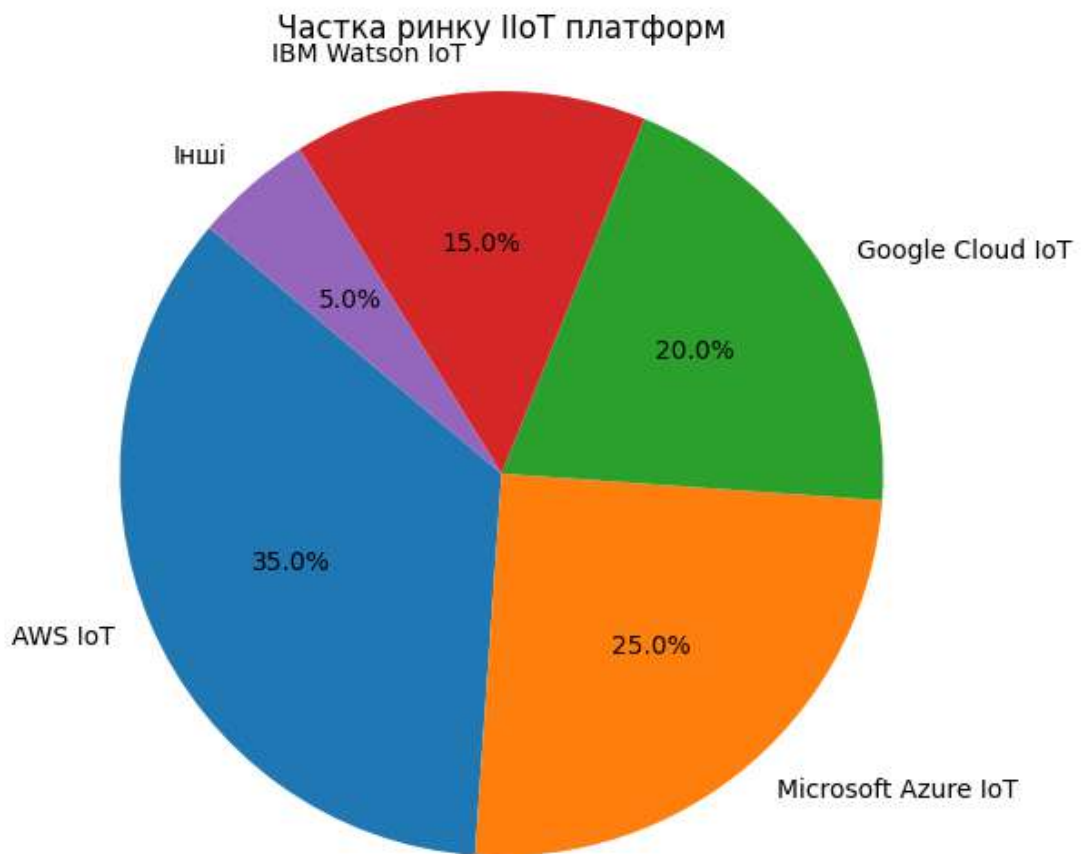


Рисунок 1.7 - Частка ринку IoT платформ

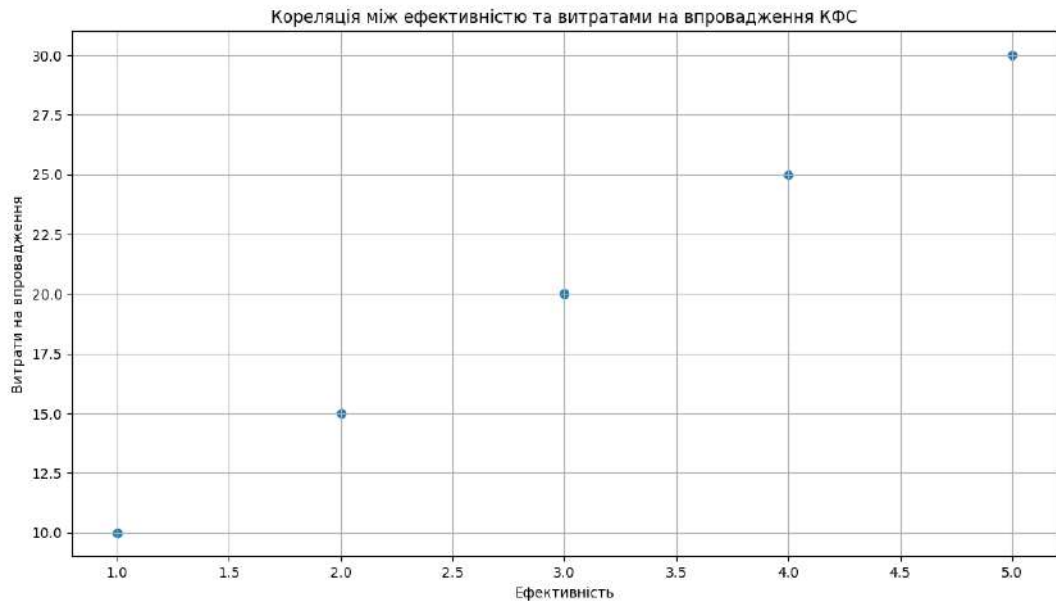


Рисунок 1.8 - Кореляція між ефективністю та витратами на впровадження КФС

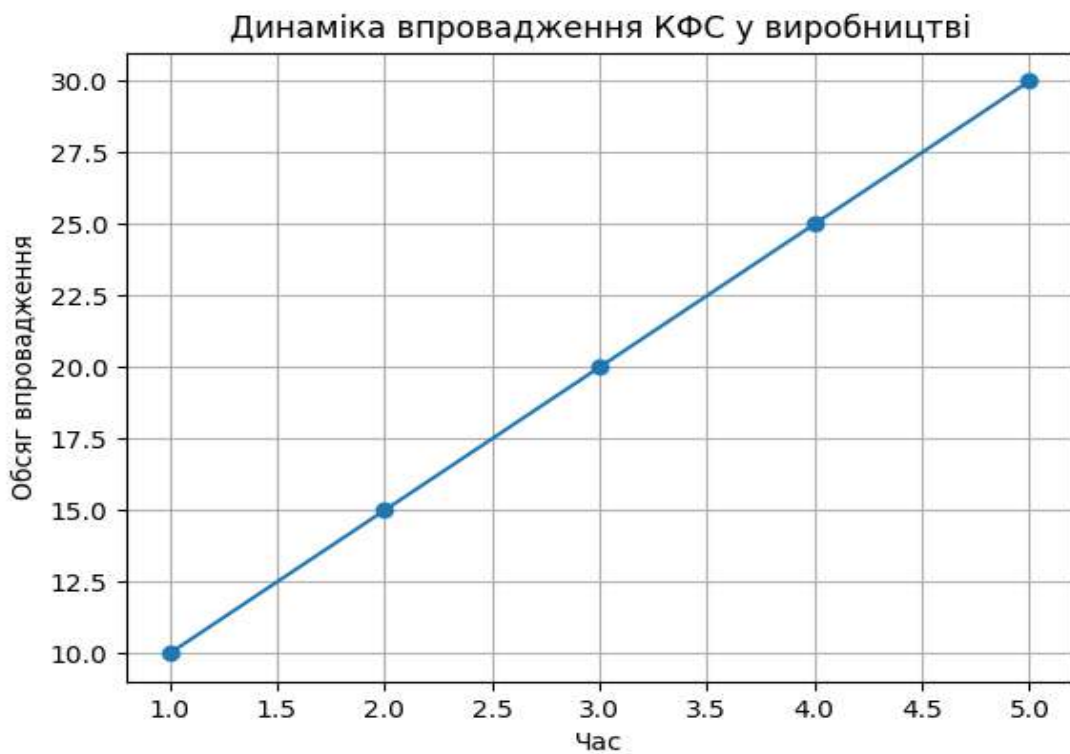


Рисунок 1.9 - Динаміка впровадження КФС у виробництві

### 1.3 Висновки до розділу

Зм.	Арк.	№ докум.	Підпис	Дата

У даному розділі було проведено детальний аналіз сучасних рішень та підходів до впровадження кіберфізичних систем у виробництві, що включає різноманітні аспекти їх використання, переваги та можливі обмеження. Кіберфізичні системи (КФС) представляють собою інтеграцію обчислювальних можливостей з фізичними процесами, що дозволяє створювати більш ефективні та оптимізовані виробничі процеси [43].

Зокрема, було розглянуто поточний стан впровадження КФС у промислових послугах виробництва, де спостерігається значний інтерес підприємств до цих технологій. Підприємства все частіше звертаються до використання кіберфізичних систем, оскільки вони здатні суттєво підвищити ефективність виробничих процесів, знизити витрати та покращити якість продукції [44].

Було проведено комплексний аналіз існуючих рішень та підходів до впровадження КФС. Зокрема, розглянуто різноманітні платформи для Індустріального Інтернету Речей (ІІІ), які дозволяють забезпечити взаємодію між різними елементами виробничих процесів, створюючи єдину інформаційну мережу. Також, особливу увагу було приділено розподіленим обчислювальним системам, що забезпечують високу продуктивність та масштабованість обробки даних [46].

Крім того, було розглянуто можливості використання технологій великих даних та штучного інтелекту у контексті кіберфізичних систем. Технології великих даних дозволяють збирати та аналізувати величезні обсяги даних, отриманих з виробничих процесів, що сприяє більш точному прийняттю рішень та підвищенню ефективності виробництва. Штучний інтелект, у свою чергу, забезпечує можливість автоматизації та оптимізації різноманітних виробничих процесів, а також передбачення можливих несправностей та їх своєчасного усунення [47].

Важливо зазначити, що кожен з цих підходів має свої переваги та обмеження, і вибір конкретного рішення повинен здійснюватися з урахуванням потреб та можливостей конкретного підприємства. Наприклад, для невеликих підприємств може бути доцільніше впровадження менш масштабних та менш витратних рішень,

тоді як великі компанії можуть дозволити собі інвестувати у більш складні та потужні системи [48].

Крім того, впровадження кіберфізичних систем (КФС) у виробництво [53] надає підприємствам можливість не лише автоматизувати та оптимізувати виробничі процеси, але й значно підвищити рівень контролю над цими процесами. Це забезпечує більш детальний моніторинг усіх етапів виробництва, дозволяє своєчасно виявляти та усувати будь-які відхилення від норми, що, в свою чергу, сприяє підвищенню ефективності виробництва та якості продукції [54]. Крім того, впровадження КФС покращує безпеку та надійність обладнання, що використовується на виробництві. Завдяки інтеграції інтелектуальних систем моніторингу та діагностики, стає можливим прогнозування можливих поломок та своєчасне проведення профілактичних заходів, що значно знижує ризик несподіваних зупинок виробництва [55].

У сучасному конкурентному середовищі ефективне використання передових технологій, таких як КФС, стає важливим фактором, що забезпечує підприємствам значні конкурентні переваги. Впровадження КФС дозволяє підприємствам не лише підвищити продуктивність та ефективність виробничих процесів, але й знизити витрати, що пов'язані з обслуговуванням обладнання та виробництвом продукції. Аналіз існуючих рішень та підходів до реалізації кіберфізичних систем у виробництві показує, що на сьогоднішній день доступний широкий спектр технологій та платформ, які можуть бути використані для впровадження КФС [56]. Серед них можна виділити платформи для Індустріального Інтернету Речей (ІІІ), розподілені обчислювальні системи, технології великих даних, штучний інтелект та інші інноваційні рішення.

Кожне з цих рішень має свої унікальні переваги та обмеження. Наприклад, платформи для ІІІ забезпечують інтеграцію різних пристроїв та систем у єдину мережу, що дозволяє ефективно збирати та аналізувати дані з різних джерел. Розподілені обчислювальні системи забезпечують високу продуктивність та масштабованість обробки даних, що є важливим для великих підприємств з великою кількістю виробничих процесів. Технології великих даних дозволяють

					КВРКІ.21/0297.21.02.10 ПЗ	Арк. 20
Зм.	Арк.	№ докум.	Підпис	Дата		

збирати та аналізувати величезні обсяги інформації, що сприяє більш точному прийняттю рішень. Штучний інтелект забезпечує можливість автоматизації та оптимізації виробничих процесів, а також передбачення можливих несправностей та їх своєчасного усунення [57].

Вибір конкретного підходу до впровадження кіберфізичних систем повинен базуватися на потребах та можливостях конкретного підприємства. Наприклад, для невеликих підприємств може бути доцільніше впровадження менш масштабних та менш витратних рішень, тоді як великі компанії можуть дозволити собі інвестувати у більш складні та потужні системи, що забезпечують максимальну ефективність та продуктивність.

Загальний висновок полягає в тому, що впровадження кіберфізичних систем у виробництво є актуальним та перспективним напрямом для оптимізації виробничих процесів [58]. Проте це вимагає уважного аналізу, вибору оптимальних рішень та врахування специфіки кожного конкретного виробництва. Отже, результати аналізу свідчать про значний потенціал використання кіберфізичних систем у виробництві, але також підкреслюють важливість уважного підходу до вибору відповідних технологій та рішень, що забезпечують максимальну ефективність та надійність виробничих процесів [59]. Це дозволить підприємствам не лише підвищити свою конкурентоспроможність, але й забезпечити стабільний та стійкий розвиток у довгостроковій перспективі [60].

					КВРКІ.21/0297.21.02.10 ПЗ	Арк. 21
Зм.	Арк.	№ докум.	Підпис	Дата		

## 2 ОБҐРУНТУВАННЯ ВИБОРУ КОМПОНЕНТІВ ТА СЕРЕДОВИЩА РЕАЛІЗАЦІЇ

### 2.1 Вимоги до компонентів кіберфізичних систем

Звідний аналіз компонентів кіберфізичних систем (КФС) виробництва, таких як сенсори, актуатори, обчислювальні ресурси та платформи, є ключовим етапом у визначенні ефективного рішення для інтеграції цих систем у виробничі процеси. Сенсори та актуатори відіграють важливу роль у виробничих системах, дозволяючи отримувати дані з фізичного середовища та керувати різними пристроями і механізмами.

Аналіз потреб виробництва показав, що система повинна підтримувати різні типи сенсорів для збору даних про умови виробництва, включаючи температурні, вологісні та тискові сенсори. Необхідно мати можливість керування різноманітними пристроями та механізмами за допомогою актуаторів, які можуть взаємодіяти зі зібраними даними та впливати на виробничий процес. Розроблена система має забезпечити ефективний збір, зберігання та аналіз даних, а також можливість відображення важливої інформації для операторів та адміністраторів. Враховуючи чутливість даних про виробничий процес, система повинна мати вбудовані механізми захисту від несанкціонованого доступу та забезпечення конфіденційності та цілісності даних.

Для побудови кіберфізичної системи в промисловому виробництві необхідно врахувати ряд вимог до компонентів системи.

#### 2.1.1 Сенсори і актуатори

Сенсори призначені для збору даних про фізичні параметри у виробничому середовищі, такі як температура, вологість, тиск.

Існує багато типів сенсорів, які можуть бути використані для збору даних у виробничому середовищі: термосенсори (вимірюють температуру та зазвичай можуть бути розміщені в середині пристрою або навіть безпосередньо в контакті з

					КВРКІ.21/0297.21.02.10 ПЗ	Арк. 22
Зм.	Арк.	№ докум.	Підпис	Дата		

робочим матеріалом), датчики вологості (вимірюють вологість повітря або матеріалу у виробничому приміщенні та можуть бути оптичними, резистивними або конденсаторними), датчики тиску (використовуються для вимірювання тиску газів або рідин у системах виробництва, можуть бути п'єзоелектричними, п'єзорезистивними або п'єзокерамічними), датчики рівня (використовуються для вимірювання рівня рідини або твердого матеріалу у резервуарах, баках або контейнерах), датчики руху (виявляють рух об'єктів або людей у виробничому приміщенні і можуть бути використані для безпеки або контролю процесів), датчики газу (вимірюють концентрацію різних газів у повітрі та можуть використовуватися для виявлення витоків або контролю рівня газів у виробничому середовищі).

В рамках дипломної роботи з кіберфізичних систем промислових послуг у виробництві обрано сенсори, які найкраще відповідають вимогам проекту. Опис використання сенсорів для проектування подано в Таблиці 2.1.

Таблиця 2.1 – Використання сенсорів для проектування

№	Тип сенсора	Назва	Опис
1	Сенсор вологості	DS18B20	Цифровий термометр, висока точність ( $\pm 0.5^{\circ}\text{C}$ ), може працювати в широкому діапазоні температур ( $-55^{\circ}\text{C}$ до $+125^{\circ}\text{C}$ ).
2	Температурні сенсори	DHT22	Комбінований датчик температури та вологості, висока точність ( $\pm 2\%$ вологості, $\pm 0.5^{\circ}\text{C}$ температури).
3	Датчики тиску	BMP280	Цифровий сенсор атмосферного тиску, висока точність ( $\pm 1.0$ hPa), може вимірювати температуру.

Ці сенсори є популярними в промисловому виробництві, мають досить високу точність та надійність, і можуть працювати в широкому діапазоні умов.

					КВРКІ.21/0297.21.02.10 ПЗ	Арк. 23
Зм.	Арк.	№ докум.	Підпис	Дата		

Однак перед вибором конкретного сенсора, рекомендується провести додаткове дослідження щодо вимог вашого проекту та особливостей виробничого середовища.

Актуатори також призначені для керування пристроями та механізмами у виробничому процесі.

Актуатори є невід'ємною складовою будь-якої кіберфізичної системи (КФС), забезпечуючи її здатність до реагування та взаємодії з фізичним середовищем. Ці пристрої відіграють критичну роль у виконанні конкретних дій або рухів на основі отриманих сигналів або даних з контролерів, комп'ютерів або навіть безпосередньо від сенсорів.

Одним з ключових аспектів актуаторів є їхнє різноманіття та адаптивність до різних завдань та потреб.

Вони можуть виконувати різноманітні фізичні дії, включаючи рух, стискання, обертання, підйом, опускання та багато інших, в залежності від конкретного застосування. Деякі актуатори спроможні працювати з великими навантаженнями або в умовах високої швидкості, що робить їх ефективними в різноманітних сценаріях виробничого середовища.

Однією з ключових переваг використання актуаторів є їхній внесок у автоматизацію процесів.

Значна увага також приділяється вдосконаленню технологій актуаторів для забезпечення їхньої енергоефективності, надійності та довговічності. Це дозволяє знижувати споживання енергії та забезпечувати стабільну та безперебійну роботу систем у протязі тривалого періоду часу.

Актуатори є невід'ємною частиною кіберфізичних систем (КФС), оскільки вони забезпечують здатність цих систем до взаємодії з реальним світом. Завдяки актуаторам, КФС можуть виконувати дії на основі отриманих даних, що робить їх ключовим компонентом в сучасних системах автоматизації та інтегрованих технологіях.

Завдяки актуаторам, кіберфізичні системи можуть виконувати широкий спектр функцій, від простих механічних рухів до складних виробничих процесів.

					КВРКІ.21/0297.21.02.10 ПЗ	Арк. 24
Зм.	Арк.	№ докум.	Підпис	Дата		

Це включає управління промисловими роботами, контроль температури, регулювання тиску, переміщення матеріалів, відкривання та закривання клапанів, а також багато інших операцій, що забезпечують ефективність та безпеку сучасних виробничих процесів.

У сучасних системах автоматизації, актуатори інтегровані з сенсорами та контролерами, що дозволяє створювати високоефективні та адаптивні системи. Інтеграція актуаторів у КФС також сприяє підвищенню точності і надійності виконання завдань, зменшуючи ймовірність помилок та підвищуючи загальну продуктивність системи. Опис використання актуаторів для проектування подано в таблиці 2.2.

Таблиця 2.2 – Використання актуаторів для проектування

Тип актуатора	Назва	Особливості
Електромотор	ДС-мотор	Використовується для обертання механізмів
Гідравлічний	Гідравлічний циліндр	Застосовується для створення сили у виробничих процесах
Пневматичний	Пневмоциліндр	Використовує стиснений повітря для виконання руху
Електромеханічні реле	5V або 12V	Напруга керування, Вихідна потужність відповідно до потреб керованих пристроїв, Кількість керованих каналів відповідно до потреб
Контролери з релею	Цифрові входи/виходи, Ethernet або Wi-Fi,	Можливість програмного керування через мережу Ethernet або Wi-Fi Надійність та безпека роботи в промисловому середовищі

## 2.1.2 Вимоги до точності вимірювань, швидкості реакції, надійності

Специфікація вимог до проекту подана в таблиці 2.3.

Таблиця 2.3 – Специфікація вимог до проекту

№	Вимога	Опис
1	Точність вимірювання	Для температурних сенсорів: точність вимірювання може бути визначена, наприклад, $\pm 0.5^{\circ}\text{C}$ або краще Для сенсорів вологості: точність вимірювання може бути вказана, наприклад, $\pm 2\%$ вологості або краще точність може бути вказана, наприклад, $\pm 1.0 \text{ hPa}$ або краще
2	Швидкість реакції	можуть варіюватися від декількох мілісекунд до кількох секунд, в залежності від конкретного застосування
3	Надійність	повинні бути надійними та безперебійно працювати протягом тривалого періоду часу без відмов вимірюється за допомогою показників, таких як середній час безвідмовної роботи (MTBF), середній час відновлення (MTTR) та інші.
4	Здатність до роботи в умовах виробництва	робота при певних температурних діапазонах, вологості, впливі вібрацій, ударів та хімічних агентів. Наприклад, датчики повинні бути стійкими до вологи та пилу, а також мати захист від викидів виробничих газів.

### 2.1.3 Мережеві пристрої

Мережеве обладнання призначене для забезпечення зв'язку та обміну даними між сенсорами, актуаторами та обчислювальними вузлами.

Мережеве обладнання в контексті кіберфізичних систем у виробництві відіграє ключову роль у забезпеченні зв'язку та обміну даними між різними компонентами системи, такими як сенсори, актуатори та обчислювальні вузли.

Таблиця 2.4 – Специфікація мережевих вимог до проекту

Вимога	Опис
Підтримка Ethernet та Wi-Fi	повинно підтримувати як провідне, так і бездротове підключення.
Надійність	повинно бути надійним та стабільним у роботі, щоб забезпечити безперебійну роботу мережі.
Швидкість реакції	повинно мати високу швидкість реакції на запити та передавати дані ефективно.
Захист даних	повинно забезпечувати захист даних у мережі та зменшувало загрози кібербезпеки.
Масштабованість	повинно бути гнучким та масштабованим для впровадження зростаючих потреб мережі.

2.1.4 Вимоги до пропускну здатності, надійності та забезпечення захисту мережі від кібератак.

Специфікація вимог пропускну здатності, надійності та забезпечення захисту до проекту подано в таблиці 2.5.

Таблиця 2.5 – Специфікація вимог пропускної здатності, надійності та забезпечення захисту до проекту

	Вимога	Опис
1	Пропускна здатність	Обладнання повинно мати достатню пропускну здатність для обробки великого обсягу даних..
2	Надійність	Мережеве обладнання має бути надійним і забезпечувати безперебійну роботу мережі.
3	Захист від кібератак	Потрібні вбудовані механізми захисту, такі як брандмауери, VPN, системи виявлення вторгнень..

### 2.1.5 Обчислювальні ресурси

Специфікація обчислювальних вузлів для обробки та аналізу даних до проекту подано в таблиці 2.6. Додаткові вимоги до проекту подано в таблиці 2.6.

Таблиця 2.6 – Специфікація обчислювальних вузлів для обробки та аналізу даних до проекту

Вимога	Опис
Висока продуктивність	Обладнання повинно мати достатню пропускну здатність для обробки великого обсягу даних..
Масштабованість	Система повинна бути здатна масштабуватися відповідно до зростання обсягів даних та завдань обробки.
Надійність	Обчислювальні вузли мають працювати безперебійно та надійно, щоб уникнути втрати даних або простою системи.
Забезпечення безпеки	Потрібні вбудовані механізми захисту, такі як шифрування даних та захист від несанкціонованого доступу.
Швидкодія	Важлива швидкодія в обчислювальних операціях, щоб забезпечити ефективну реакцію на зміни в системі.

Таблиця 2.7 – Додаткові вимоги до проекту

	Вимога	Опис
1	Висока продуктивність	Обчислювальні вузли повинні мати достатню продуктивність для ефективної обробки даних
2	Швидкість обробки даних	Система повинна забезпечувати швидку обробку великих обсягів даних без затримок
3	Великий обсяг пам'яті	Обчислювальні вузли повинні мати достатній обсяг оперативної та постійної пам'яті
4	Забезпечення безпеки	Мають бути вбудовані механізми безпеки для захисту даних та забезпечення конфіденційності.

### 2.1.6 Програмне забезпечення

Додаткові вимоги програмного забезпечення до проекту подано в таблиці 2.8.

Таблиця 2.8 – Додаткові вимоги програмного забезпечення до проекту

	Вимога	Опис
1	Управління системою	Програмне забезпечення повинно забезпечувати можливість керування різними аспектами кіберфізичної системи, такими як включення та виключення пристроїв, налаштування параметрів роботи сенсорів та актуаторів, управління мережевими з'єднаннями тощо.
2	Моніторинг даних	Програмне забезпечення повинно забезпечити можливість моніторингу важливих параметрів системи, таких як температура, вологість, тиск, стан пристроїв та зв'язків. Це допомагає операторам виробництва отримувати реальний час інформації про стан обладнання та виробничих процесів.

Кінець таблиці 2.8 – Додаткові вимоги програмного забезпечення до проекту

	Вимога	Опис
3	Збір та аналіз даних	Програмне забезпечення повинно мати можливості для збору, зберігання та аналізу даних з різних датчиків та пристроїв у реальному часі. Це допомагає виявляти тенденції, попереджати про можливі несправності або аварії та робити прийняття рішень на основі аналітичних даних.
4	Інтеграція з іншими системами	Програмне забезпечення повинно бути здатним інтегруватися з іншими системами у виробничому середовищі, такими як системи управління виробництвом (MES), системи планування ресурсів підприємства (ERP) або системи автоматизації обладнання.

2.1.7 Додаткове програмне забезпечення для аналізу даних, взаємодії з користувачем та вирішення конкретних завдань виробництва

Вимоги до додатковго програмного забезпечення подано в таблиці 2.9.

Таблиця 2.9 – Вимоги до додатковго програмного забезпечення

	Вимога	Опис
1	Підтримка різноманітних форматів даних	Програмне забезпечення повинно бути здатним працювати з різними форматами даних, включаючи структуровані та неструктуровані дані з різних джерел
2	Аналітичні можливості	Потрібно мати можливість виконувати різні види аналізу даних, включаючи статистичний аналіз, машинне навчання, аналіз великих даних та візуалізацію результатів.

Кінець тблиці 2.9 – Вимоги до додатковго програмного забезпечення

	Вимога	Опис
3	Інтеграція з існуючими системами	Програмне забезпечення повинно бути легко інтегровано з існуючими системами автоматизації виробництва та має підтримувати обмін даними між ними.
4	Можливості взаємодії з користувачем:	Потрібно мати інтерфейс користувача, який дозволить операторам легко взаємодіяти з програмним забезпеченням, вводити параметри, переглядати результати аналізу та приймати рішення на основі цих даних.
5	Захист даних	Програмне забезпечення повинно забезпечувати захист конфіденційності та цілісності даних, включаючи механізми аутентифікації, авторизації та шифрування.

1.2.8 Вимоги до безпеки, надійності та сумісності з іншими компонентами системи.

Необхідно забезпечити захист від несанкціонованого доступу, атак з метою витоку даних або порушення функціональності системи, стабільної та неперервної роботи системи без відмов, а також здатність програмного забезпечення працювати відповідно до встановлених стандартів та взаємодіяти з іншими компонентами системи.

Механізми захисту: використання сучасних механізмів захисту, таких як брандмауери, системи виявлення вторгнень (IDS), системи захисту від вторгнень (IPS) і антивірусне програмне забезпечення.

Шифрування даних відбувається із застосуванням сильних алгоритмів шифрування для захисту конфіденційності даних під час їх передачі та зберігання. Потрібно використання методів аутентифікації, таких як паролі, біометричні дані

або двофакторна аутентифікація, для перевірки ідентичності користувачів та забезпечення доступу тільки авторизованим особам.

Для забезпечення масштабованості системи необхідно враховувати ряд ключових аспектів, що гарантують її гнучкість та здатність легко адаптуватися до змін у виробничих потребах. По-перше, система повинна мати модульну архітектуру, що дозволяє додавати нові компоненти або модифікувати існуючі без необхідності повного перепроєктування всієї системи. Це забезпечує високу гнучкість, дозволяючи системі адаптуватися до нових викликів та потреб виробництва без значних витрат часу та ресурсів.

Архітектура системи повинна бути складною, але водночас зрозумілою для інженерів та операторів, які її експлуатують. Це означає, що всі компоненти системи повинні бути чітко документовані, а їх взаємодія – легко зрозумілою. Такий підхід дозволяє швидко і безпомилково проводити розширення та модернізацію системи, що є критично важливим для підтримання її продуктивності та надійності.

Крім того, система повинна бути здатною обробляти великі обсяги даних, які постійно зростають з часом.

Для цього необхідно забезпечити ефективні механізми збору, зберігання та обробки даних.

Важливо, щоб система надавала швидкий доступ до цих даних для подальшого аналізу та прийняття обґрунтованих рішень. Це може включати використання розподілених обчислювальних ресурсів та технологій великих даних для забезпечення швидкої та ефективної обробки інформації.

Не менш важливою є наявність механізмів резервного копіювання та відновлення даних. Система повинна забезпечувати безперервну роботу навіть у разі виникнення проблем або відмови обладнання.

Це включає створення регулярних резервних копій даних та розробку планів відновлення, які дозволяють швидко відновити роботу системи після аварійних ситуацій. Такий підхід мінімізує ризик втрати даних та забезпечує високу надійність системи.

					КВРКІ.21/0297.21.02.10 ПЗ	Арк. 32
Зм.	Арк.	№ докум.	Підпис	Дата		

Для забезпечення ефективного використання ресурсів система повинна підтримувати як горизонтальне, так і вертикальне масштабування.

Горизонтальне масштабування передбачає додавання нових серверів та обчислювальних ресурсів до системи, що дозволяє збільшувати її потужність у міру зростання потреб.

Вертикальне масштабування передбачає підвищення потужності індивідуальних компонентів системи, таких як процесори, пам'ять та сховища даних, що дозволяє покращувати продуктивність системи без зміни її архітектури.

Отже, для забезпечення масштабованості системи необхідно враховувати всі ці аспекти, що дозволяють створити гнучку, надійну та продуктивну систему, яка здатна ефективно адаптуватися до змін у виробничих потребах.

Це забезпечить підприємствам можливість максимально ефективно використовувати свої ресурси, знижуючи витрати та підвищуючи продуктивність виробничих процесів.

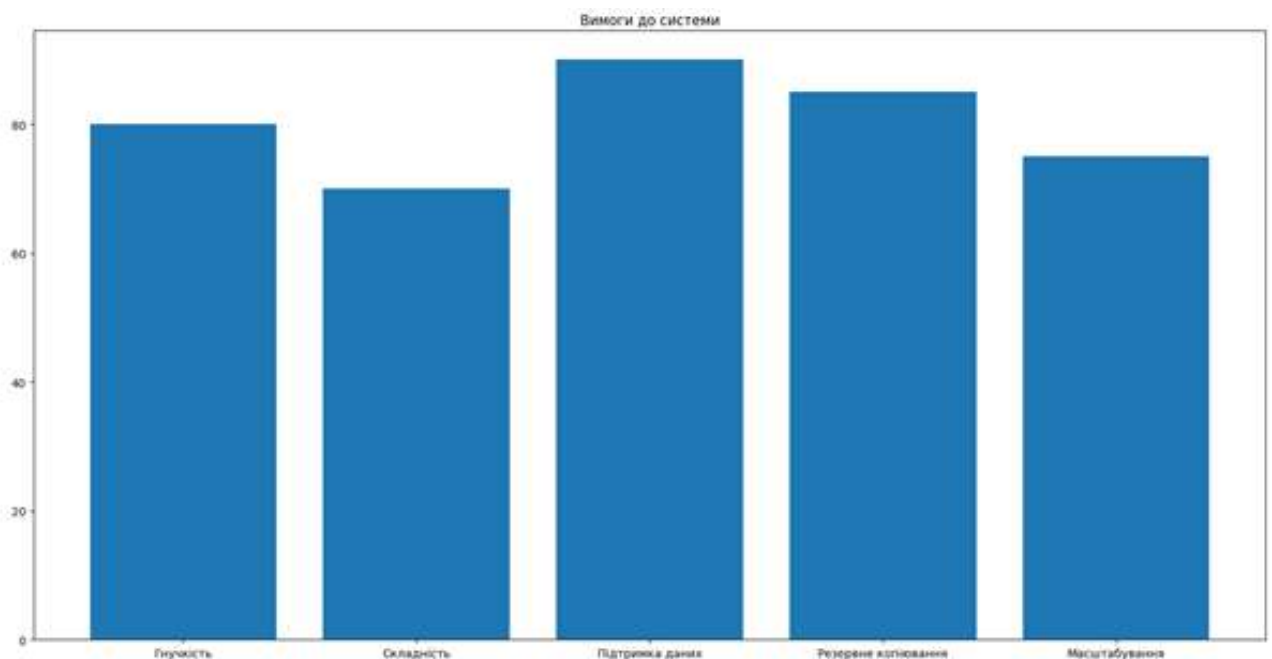


Рисунок 2.1 – Вимоги до системи

## 2.2 Вибір середовища реалізації

### 2.2.1 Огляд можливих середовищ для реалізації КФС

Для впровадження кіберфізичних систем (КФС) виробництва необхідно вибрати середовище, яке відповідає вимогам проекту та забезпечує ефективну реалізацію системи. Огляд деяких можливих середовищ:

1. Хмарні платформи, такі як Amazon Web Services (AWS), Microsoft Azure та Google Cloud Platform (GCP), надають широкий набір інструментів та послуг для розгортання, керування та масштабування КФС. Вони забезпечують високу доступність, масштабованість, безпеку, а також гнучкість у використанні ресурсів. Наприклад, AWS пропонує такі сервіси як AWS IoT Core для керування пристроями та збирання даних, а також AWS Greengrass для обробки даних на межі мережі. Microsoft Azure має Azure IoT Hub, що дозволяє легко інтегрувати різні IoT пристрої та керувати ними, тоді як GCP пропонує Google IoT Core, що дозволяє підключати та керувати пристроями, а також обробляти їхні дані за допомогою інших хмарних сервісів.

2. Використання власних локальних серверів у виробничому приміщенні може бути вигідним з точки зору контролю та безпеки даних, проте воно вимагає більших інвестицій у придбання та обслуговування обладнання, а також у розробку та підтримку відповідної інфраструктури. Локальні сервери дозволяють забезпечити мінімальні затримки та високу швидкість обробки даних, що є критичним для деяких виробничих процесів.

3. Використання обробки на межі мережі дозволяє обробляти дані ближче до джерела їх виникнення, що зменшує затримки та навантаження на мережу. Це особливо корисно для додатків, які потребують роботи в реальному часі. Обробка на межі мережі також може покращити безпеку та зменшити залежність від хмарних сервісів. Приклади платформ для edge computing включають AWS Greengrass, Azure IoT Edge та Google Edge TPU.

4. Платформи IoT, такі як IBM Watson IoT, AWS IoT Core та Azure IoT Hub, спеціалізовані на зборі, обробці та аналізі даних в реальному часі з IoT пристроїв.

					КВРКІ.21/0297.21.02.10 ПЗ	Арк. 34
Зм.	Арк.	№ докум.	Підпис	Дата		

Вони надають інструменти для підключення різноманітних сенсорів та пристроїв, збору великих обсягів даних, їхньої обробки та аналізу, а також інтеграції з іншими системами. Наприклад, IBM Watson IoT використовує технології штучного інтелекту для аналізу даних та виявлення аномалій, що допомагає оптимізувати виробничі процеси.

У багатьох випадках оптимальне середовище для впровадження КФС – це комбінація різних платформ та технологій. Наприклад, дані можуть бути зібрані на мікроконтролерах на місці, оброблені на межі мережі для зменшення затримок, а потім відправлені на хмарне середовище для подальшого аналізу та зберігання. Такий підхід дозволяє максимально ефективно використовувати переваги кожної з технологій, забезпечуючи гнучкість, надійність та високу продуктивність кіберфізичних систем.

### 2.2.2 Огляд можливих середовищ для розробки

Python є не лише популярним, але й надійним вибором завдяки своїм властивостям простоти в освоєнні, гнучкості та широкому спектру наявних бібліотек для обробки даних, машинного навчання та інших аспектів аналізу та обробки даних в кіберфізичних системах. Ця мова програмування стала популярною серед спеціалістів у багатьох галузях, включаючи розробку програмного забезпечення для КФС, завдяки своїй простоті в написанні коду та широким можливостям вирішення різних завдань.

Однак вибір середовища розробки залежить від конкретних потреб проекту та уподобань розробника. На сьогоднішній день існують десятки середовищ розробки для Python, кожне з яких має свої переваги та недоліки. Наприклад, Visual Studio Code відомий своєю легкістю використання та розширюваності, PyCharm відрізняється великою кількістю інтегрованих функцій та інтелектуальних інструментів, а Jupyter Notebook забезпечує можливість інтерактивного програмування та візуалізації даних.

Крім того, зростанням популярності Python у сфері КФС з'являються нові інструменти та середовища розробки, спеціально призначені для роботи з цими системами. Вибір конкретного середовища може бути підтриманий обґрунтованою оцінкою вимог проекту, особливостей команди розробників та доступних ресурсів.

Visual Studio Code (VS Code) - це не просто надзвичайно зручне та потужне середовище розробки, а справжній мультифункціональний інструмент, що відкриває безліч можливостей для створення програмного забезпечення для кіберфізичних систем (КФС). Він володіє безліччю корисних функцій та розширень, що роблять його найбільш впізнаваним та вибраним середовищем для розробки програмного забезпечення у різних галузях, включаючи КФС.

Однією з найважливіших переваг VS Code є його універсальність та підтримка різноманітних мов програмування. Незалежно від того, чи ви працюєте з Python, C++, JavaScript, або іншими мовами, VS Code надає вам потужні інструменти та зручний інтерфейс для продуктивної роботи. Ви можете легко налаштувати його під свої потреби за допомогою розширень, що робить робочий процес більш гнучким та ефективним.

Загалом, Visual Studio Code не лише допомагає вам створювати програмне забезпечення для КФС з високою ефективністю та продуктивністю, але також забезпечує надійне та зручне середовище для вашої роботи, дозволяючи вам фокусуватися на творчості та досягненні ваших цілей розробки.

PyCharm - це інтегроване середовище розробки (IDE) для мови програмування Python, яке забезпечує розробникам зручні інструменти для створення різноманітного програмного забезпечення. Це середовище розробки особливо корисне для створення програмного забезпечення для кіберфізичних систем, оскільки надає широкий спектр функцій, що полегшують процес розробки, тестування та налагодження коду. PyCharm підтримує різні мови програмування, включаючи Python, JavaScript, HTML, CSS, і це робить його універсальним інструментом для розробників, які працюють над проектами різного типу та складності.

Однією з ключових переваг PyCharm є його можливості для розробки веб-додатків. Завдяки підтримці таких мов, як JavaScript, HTML, і CSS, розробники можуть створювати повноцінні веб-додатки для кіберфізичних систем, що забезпечують інтерактивний та зручний користувацький інтерфейс. Це особливо важливо для кіберфізичних систем, де користувачі повинні мати можливість легко взаємодіяти з системою через веб-інтерфейс, відстежувати параметри, налаштовувати обладнання та отримувати інформацію про стан системи.

PyCharm доступний для різних операційних систем, включаючи Windows, macOS та Linux. Це забезпечує гнучкість для розробників, дозволяючи їм працювати в знайомому середовищі незалежно від того, яку операційну систему вони використовують. Підтримка кількох платформ робить PyCharm зручним вибором для команд розробників, де учасники можуть працювати на різних операційних системах, але використовувати одне інтегроване середовище розробки для всіх своїх проєктів.

Крім того, PyCharm має багато корисних функцій, які підвищують продуктивність розробників. Серед них – інтелектуальне автодоповнення коду, потужні інструменти для рефакторингу, інтеграція з системами контролю версій, такими як Git, підтримка тестування, відладка коду в режимі реального часу та багато інших. Ці функції допомагають розробникам швидше писати код, уникати помилок і легко знаходити та виправляти їх, що особливо важливо для складних проєктів. PyCharm також підтримує інтеграцію з різними фреймворками та бібліотеками для Python, такими як Django, Flask, NumPy, Pandas та інші. Це дозволяє розробникам використовувати всі переваги цих інструментів безпосередньо в середовищі розробки, що робить процес розробки більш ефективним та зручним. Наприклад, розробники можуть легко створювати веб-додатки з використанням Django або аналізувати дані з допомогою Pandas, не покидаючи PyCharm.

Eclipse - це відкрите інтегроване середовище розробки (IDE), яке використовується для розробки різноманітного програмного забезпечення, включаючи, зокрема, кіберфізичні системи. Це середовище надає розробникам

					КВРКІ.21/0297.21.02.10 ПЗ	Арк. 37
Зм.	Арк.	№ докум.	Підпис	Дата		

потужний набір інструментів для створення, налагодження та тестування програм, забезпечуючи високий рівень гнучкості та функціональності. Завдяки своїй відкритій архітектурі, Eclipse підтримує велику кількість плагінів, які можна додатково встановлювати для розширення можливостей середовища. Це дозволяє розробникам налаштовувати Eclipse відповідно до специфічних потреб своїх проектів і забезпечує широкий спектр можливостей для розробки програмного забезпечення різного типу.

Крім того, Eclipse підтримує безліч мов програмування, таких як Java, C++, Python, PHP та інші. Це робить його універсальним інструментом для розробки програмного забезпечення в різних галузях. Розробники можуть використовувати Eclipse для створення настільних, веб- та мобільних додатків, а також для роботи з базами даних і розробки вбудованих систем. Завдяки широкому спектру інструментів та підтримці різних мов програмування, Eclipse стає незамінним інструментом для багатьох розробників.

На додаток до Eclipse, розробники часто використовують ще одне популярне інтегроване середовище розробки – IntelliJ IDEA. IntelliJ IDEA підтримує різні мови програмування, включаючи Java, Kotlin, Groovy, Scala, JavaScript, TypeScript, PHP, Python та багато інших. Це дозволяє розробникам працювати над проектами різного типу та масштабу, використовуючи єдине середовище. IntelliJ IDEA надає потужний набір інструментів для написання, рефакторингу, налагодження та тестування коду, що робить процес розробки більш ефективним та зручним.

IntelliJ IDEA також підтримує інтеграцію з різними системами контролю версій, такими як Git, Mercurial та SVN, що дозволяє розробникам легко керувати своїм кодом і співпрацювати з іншими членами команди. Крім того, IntelliJ IDEA має вбудовану підтримку для роботи з різними фреймворками та технологіями, такими як Spring, Hibernate, Java EE, React, Angular та багатьма іншими. Це робить його універсальним інструментом для розробки сучасних додатків.

Jupyter Notebook - це потужне інтерактивне середовище розробки, яке надає розробникам можливість створювати та виконувати код у зручному блокнотному форматі. Це середовище є особливо корисним для наукових досліджень, аналізу

					КВРКІ.21/0297.21.02.10 ПЗ	Арк. 38
Зм.	Арк.	№ докум.	Підпис	Дата		

даних, машинного навчання та освіти, оскільки дозволяє інтерактивно працювати з кодом, даними та результатами обчислень у єдиному документі. Однією з основних переваг Jupyter Notebook є його підтримка різних мов програмування, включаючи, але не обмежуючись, Python, R, та Julia. Це дозволяє розробникам працювати в одному середовищі, навіть якщо проекти вимагають використання кількох мов програмування.

Крім написання та виконання коду, Jupyter Notebook також підтримує додавання текстових блоків, що використовують синтаксис Markdown. Це дає можливість документувати процес розробки, пояснювати кроки обчислень, додавати заголовки, списки, посилання, зображення та інші елементи форматування безпосередньо в нотатки. Використання Markdown дозволяє створювати добре структуровані та легко читані документи, які можуть бути корисними як для особистих записів, так і для спільної роботи в команді.

Jupyter Notebook інтегрується з багатьма іншими інструментами та сервісами, що робить його ще більш гнучким та корисним. Наприклад, він може працювати разом з бібліотеками для обробки та аналізу даних, такими як Pandas, NumPy, Matplotlib, Seaborn та іншими, що дозволяє легко завантажувати, обробляти, аналізувати та візуалізувати дані.

Ще однією важливою функцією Jupyter Notebook є можливість створення інтерактивних візуалізацій. Завдяки підтримці бібліотек, таких як Plotly, Bokeh, Altair та інші, користувачі можуть створювати інтерактивні графіки та візуалізації, які допомагають краще зрозуміти дані та представити результати аналізу в зрозумілій та наочній формі.

## 2.2.2 Порівняння та обґрунтування вибору конкретного середовища

В Табл. 2.10 показано загальний огляд різних середовищ розробки з урахуванням їхніх основних функцій та можливостей. Вибір конкретного середовища залежить від потреб та особливостей вашого проекту. На рис. 2.2.2

					КВРКІ.21/0297.21.02.10 ПЗ	Арк. 39
Зм.	Арк.	№ докум.	Підпис	Дата		

відображено вибір мови програмування та середовища розробки для кіберфізичних систем.

Таблиця 2.10 – Порівняльна характеристика між різними середовищами розробки

№	Функція	VS Code	PyCharm	Eclipse	IntelliJ IDEA	Jupyter Notebook
1	Мови програмування	40+	1	40+	30+	50+
2	Відлагодження	Так	Так	Так	Так	Ні
3	Автодоповнення	Так	Так	Так	Так	Ні
4	Інтеграція з Git	Так	Так	Так	Так	Так
5	Відкритий код	Так	Ні	Так	Ні	Так
6	Вбудовані інструменти	Так	Так	Так		Ні
7	Використання ресурсів	Низьке	Високе	Високе	Високе	Низьке

Вибір конкретного середовища розробки Visual Studio Code (VS Code), може бути обґрунтований кількома факторами:

1. Має простий та інтуїтивно зрозумілий інтерфейс користувача, що полегшує швидку розробку та налагодження коду.
2. Підтримується на багатьох платформах, включаючи Windows, macOS і Linux, що робить його доступним для розробників на різних операційних системах.
3. Має розширену екосистему розширень, яка дозволяє розширити його функціональність для підтримки різних мов програмування, рамок розробки, інструментів та інших функцій.
4. Має велику кількість розширень, що дозволяють розширити його функціональність для підтримки різних мов програмування, інструментів та технологій, які часто використовуються в кіберфізичних системах.

5. Має вбудовану підтримку для Git, що дозволяє зручно працювати з системами контролю версій безпосередньо з редактора.

6. Має багато розширень та функцій, що полегшують роботу з цією мовою програмування.

7. Має можливість інтеграції з різними хмарними сервісами, такими як Azure, AWS, Google Cloud, що полегшує розробку хмарних додатків та послуг.

Особливості між мовами програмування та середовища розробки подано на рисунку 2.2.

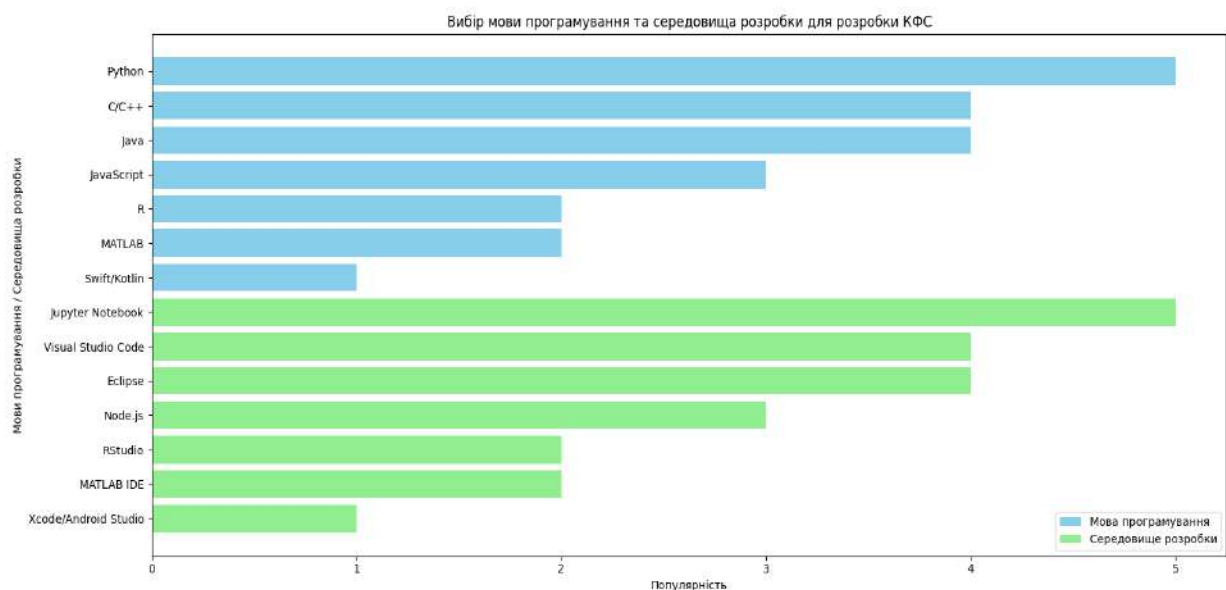


Рисунок 2.2 – Особливості між мовами програмування та середовища розробки

### 2.3 Забезпечення безпеки та масштабованості кіберфізичних систем

Огляд архітектурних підходів для автоматизованих систем є ключовим етапом у проектуванні ефективної та надійної системи. Враховуючи складність та різноманітність сучасних технологій, цей процес потребує детального аналізу та всебічного розгляду різних аспектів безпеки.

Першим кроком у цьому процесі є проведення всебічного аналізу загроз безпеки, що включає виявлення потенційних вразливостей системи та визначення

можливих шляхів атак. Такий аналіз дозволяє зрозуміти, які компоненти системи найбільше вразливі до кібератак і які заходи необхідно вжити для їх захисту. Наприклад, віддалений доступ до даних через мережу може бути використаний для несанкціонованого зчитування конфіденційної інформації про виробничі процеси. Це може призвести до значних збитків для підприємства, включаючи втрату комерційної таємниці або компрометацію виробничих процесів.

Одним з основних заходів для забезпечення безпеки є встановлення захисту мережі за допомогою брандмауера (фаєрволу) та системи виявлення вторгнень (IDS/IPS). Брандмауер допомагає контролювати вхідний та вихідний трафік, дозволяючи лише санкціоновані з'єднання, тоді як система виявлення вторгнень моніторить мережеву активність та виявляє потенційні атаки. Наприклад, налаштування брандмауера може включати фільтрацію IP-адрес, обмеження доступу до певних портів та застосування правил для захисту від відомих типів атак.

Важливим аспектом захисту даних є шифрування трафіку між компонентами системи. Використання протоколів TLS/SSL забезпечує захист даних під час їх передачі через мережу, роблячи їх недоступними для злоумисників. Це особливо важливо для систем, які обробляють конфіденційну інформацію або працюють у режимі реального часу. Наприклад, шифрування дозволяє захистити дані від перехоплення під час їх передачі між сервером та клієнтом, зменшуючи ризик несанкціонованого доступу.

Регулярне оновлення програмного забезпечення є критично важливим для підтримки безпеки системи. Використання офіційних патчів безпеки допомагає вирішувати виявлені вразливості та запобігати можливим атакам. Програмне забезпечення без регулярних оновлень може містити відомі вразливості, які злоумисники можуть використовувати для проникнення в систему. Наприклад, оновлення операційної системи, застосунків та антивірусного програмного забезпечення дозволяє захистити систему від нових типів загроз.

Для забезпечення постійного контролю за станом безпеки необхідно налаштувати систему моніторингу та аудиту, яка автоматично виявлятиме та

					КВРКІ.21/0297.21.02.10 ПЗ	Арк. 42
Зм.	Арк.	№ докум.	Підпис	Дата		

реагуватиме на підозрілу активність у системі. Така система дозволяє вчасно виявляти та запобігати можливим атакам, а також проводити розслідування у випадку інцидентів безпеки. Наприклад, система моніторингу може реєструвати будь-які спроби незаконного доступу, змін у конфігурації системи або підозрілу активність, надсилаючи сповіщення адміністратору для негайного реагування. Це дозволяє швидко виявляти та усувати проблеми, забезпечуючи безперебійну роботу системи.

Регулярне технічне обслуговування системи є невід'ємною частиною забезпечення її надійної роботи та безпеки. Під час періодичних обслуговувань проводиться перевірка стану обладнання, оновлення програмного забезпечення, а також тестування системи на наявність вразливостей. Наприклад, під час обслуговування можуть бути встановлені останні оновлення безпеки, проведено аналіз журналів системи та виконано тести на проникнення, щоб виявити та усунути можливі загрози.

Узагальнюючи, процес забезпечення безпеки автоматизованих систем включає всебічний аналіз загроз, встановлення захисних заходів, регулярне оновлення програмного забезпечення, налаштування системи моніторингу та проведення періодичних технічних обслуговувань. Ці заходи допомагають забезпечити надійну та безпечну роботу системи, захищаючи її від потенційних кібератак та зловмисних дій.

Забезпечення масштабованості є критичним аспектом в розробці кіберфізичних систем (КФС). Масштабованість визначає здатність системи ефективно працювати під зростанням обсягу даних, навантаження або кількості користувачів без втрати продуктивності або функціональності. Для забезпечення масштабованості враховано наступні аспекти:

1. Горизонтальне масштабування полягає в додаванні нових ресурсів, таких як сервери або обчислювальні вузли, для розподілу навантаження між ними. Цей підхід дозволяє розподілити обробку даних та виконання завдань між декількома машинами, що зменшує навантаження на кожен окремий компонент і підвищує загальну продуктивність системи. Наприклад, у випадку зі збільшенням кількості

					КВРКІ.21/0297.21.02.10 ПЗ	Арк. 43
Зм.	Арк.	№ докум.	Підпис	Дата		

користувачів, нові сервери можуть бути додані до кластеру для обробки додаткових запитів, зменшуючи тим самим час відгуку і підвищуючи надійність системи.

2. Вертикальне масштабування передбачає збільшення потужності окремих компонентів системи, таких як обчислювальні вузли чи бази даних. Це може включати додавання оперативної пам'яті, підвищення швидкості процесорів або збільшення обсягу дискового простору. Вертикальне масштабування дозволяє збільшити продуктивність конкретного компонента без необхідності додавання нових вузлів. Наприклад, якщо база даних стає "вузьким місцем" у системі, збільшення її потужності може значно підвищити швидкість обробки запитів та зменшити затримки.

3. Розбиття системи на невеликі незалежні компоненти, які можуть масштабуватися окремо, дозволяє ефективно впоратися зі зростанням навантаження. Мікросервісна архітектура передбачає, що кожен компонент системи є автономним сервісом, який може бути розвинутий, розгорнутий та масштабований незалежно від інших. Це забезпечує гнучкість у розподіленні ресурсів та дозволяє кожному мікросервісу обслуговувати тільки свою частину функціональності, що зменшує залежність між компонентами. Наприклад, якщо один з мікросервісів, відповідальний за обробку платежів, відчуває збільшення навантаження, його можна масштабувати окремо, не впливаючи на інші частини системи.

4. Розподілення обчислювальних завдань між різними вузлами системи дозволяє використовувати ресурси ефективніше та прискорює обробку даних. Цей підхід дозволяє уникнути перевантаження окремих компонентів та забезпечити рівномірне розподілення навантаження. Наприклад, при обробці великих обсягів даних можна використовувати кластер обчислювальних вузлів, кожен з яких виконує частину завдань, що прискорює загальний процес обробки та підвищує продуктивність системи.

5. Налаштування системи для автоматичного виявлення навантаження та реакції на нього, наприклад, через автоматичне масштабування, дозволяє

					КВРКІ.21/0297.21.02.10 ПЗ	Арк. 44
Зм.	Арк.	№ докум.	Підпис	Дата		

забезпечити оптимальну продуктивність системи у всіх умовах. Автоматичне масштабування передбачає використання алгоритмів, які аналізують поточне навантаження на систему та автоматично додають або зменшують ресурси залежно від потреб. Це дозволяє системі динамічно адаптуватися до змін навантаження, забезпечуючи стабільну роботу. Наприклад, у пікові періоди, такі як святкові дні або рекламні кампанії, система може автоматично додавати нові сервери для обробки збільшеного обсягу запитів, а після завершення пікових навантажень ресурси можуть бути зменшені, щоб знизити витрати.

#### 2.4 Висновок до розділу

У розділі 2 було розглянуто вимоги до компонентів кіберфізичних систем, вибір середовища реалізації та забезпечення безпеки та масштабованості цих систем. Під час аналізу вимог до компонентів, визначено ключові параметри, такі як типи сенсорів і актуаторів, їх точність вимірювань, швидкість реакції та надійність. У розділі про вибір середовища реалізації розглянуто різні можливі середовища, такі як Visual Studio Code, PyCharm, Eclipse, IntelliJ IDEA та Jupyter Notebook, порівняли їх за різними критеріями та обґрунтували вибір Visual Studio Code. Нарешті, також розглянуто питання безпеки та масштабованості кіберфізичних систем і визначили основні вимоги до цих аспектів.

В цілому, розділ 2 надає загальний огляд та аналіз різних аспектів розробки кіберфізичних систем, що допоможе в подальшій роботі над проектом.

					КВРКІ.21/0297.21.02.10 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		45

### 3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

#### 3.1 Розробка алгоритмів для збору та аналізу даних

Для розробки алгоритмів у середовищі Python можна використовувати багато інструментів та бібліотек, що дозволяють легко працювати з даними, здійснювати взаємодію з пристроями та проводити аналіз. Деякі з цих інструментів та бібліотек включають у себе:

- 1) NumPy та Pandas - для роботи з даними та їх обробки;
- 2) Matplotlib та Seaborn - для візуалізації даних;
- 3) SciPy та scikit-learn - для використання алгоритмів машинного навчання;
- 4) TensorFlow та PyTorch - для розробки та навчання моделей глибокого навчання;
- 5) Flask або Django - для розробки веб-інтерфейсу для взаємодії з системою.

Завдяки широкому спектру інструментів та бібліотек, які надає мова програмування Python, розробка алгоритмів для збору та аналізу даних у кіберфізичних системах може бути здійснена дуже ефективно та зручно. Python пропонує різноманітні бібліотеки, такі як Pandas для обробки даних, NumPy для числових обчислень, Matplotlib і Seaborn для візуалізації даних, а також Scikit-Learn та TensorFlow для машинного навчання. Використовуючи ці інструменти, можна легко і швидко розробляти та впроваджувати складні алгоритми аналізу та прогнозування даних.

Для прикладу, було розглянуто завдання навчання моделі для прогнозування температури на основі вимірювань вологості та тиску з набору даних, що містить ці показники з різних датчиків. Припускається, що є набір даних, який включає вимірювання температури (у градусах Цельсія), вологості (у відсотках) та тиску (у гектопаскалях) за кілька днів. Таблиця 3.1.1 демонструє ці дані, де кожен рядок відповідає одному дню вимірювань. З використанням цього набору даних можна навчити модель прогнозувати температуру на основі вимірювань вологості та тиску.

Для початку, дані необхідно попередньо обробити. Це включає очищення даних від можливих аномалій та пропусків, нормалізацію та масштабування значень, що забезпечить кращу продуктивність моделі. Далі, з використанням бібліотеки Pandas, можна легко завантажити та обробити дані, виконуючи різні операції з таблицями та даними, щоб підготувати їх до аналізу.

Наступним кроком є вибір моделі машинного навчання для прогнозування температури. Для цього можна скористатися бібліотекою Scikit-Learn, яка пропонує широкий спектр алгоритмів машинного навчання, таких як лінійна регресія, випадкові ліси, градієнтний бустинг та інші. Наприклад, лінійна регресія може бути використана для створення моделі, яка буде прогнозувати температуру на основі заданих вхідних параметрів (вологості та тиску). Вибір алгоритму залежить від характеристик даних та вимог до точності прогнозу.

Навчання моделі включає поділ даних на тренувальний та тестовий набори, що дозволяє оцінити якість моделі. Після тренування моделі, її можна перевірити на тестовому наборі даних для оцінки точності та надійності прогнозів. Можливо, знадобиться провести кілька ітерацій з налаштуванням гіперпараметрів моделі для досягнення оптимальних результатів.

Після успішного навчання та тестування моделі, її можна інтегрувати у систему для автоматичного прогнозування температури на основі нових вимірювань вологості та тиску. Це дозволить оперативно отримувати прогнози температури, що може бути корисним для різних виробничих процесів та моніторингу стану систем, забезпечуючи своєчасне виявлення можливих відхилень та запобігання потенційним збоїв, таким чином підвищуючи загальну ефективність та стабільність виробництва.

Таким чином, завдяки потужним інструментам та бібліотекам Python, розробка моделей для прогнозування на основі даних з кіберфізичних систем стає простою та ефективною задачею. Використовуючи сучасні методи машинного навчання та обробки даних, можна досягти високої точності прогнозів, що сприяє підвищенню ефективності та надійності виробничих процесів, дозволяючи

					КВРКІ.21/0297.21.02.10 ПЗ	Арк. 47
Зм.	Арк.	№ докум.	Підпис	Дата		

своєчасно виявляти та виправляти відхилення, оптимізувати робочі потоки та мінімізувати ризики простоїв.

Таблиця 3.1 – Попередній набір даних з показників

Дата	Температура (°C)	Вологість (%)	Тиск (hPa)
2024-01-01	20.5	50	1013
2024-01-02	19.8	48	1012
2024-01-03	21.2	52	1014
2024-01-04	20.09	49	1011
2024-01-05	20.3	51	1010
2024-01-06	21.5	53	1012

Приклад масивів NumPy з вимірюваннями температури, вологості та тиску:

```
# Підготовка даних
# Припускається, що вже є дані у вигляді масивів NumPy
# Ці дані містять вимірювання температури, вологості та тиску
# X містить вологість та тиск, а Y містить вимірювання температури
X = np.array([[49, 1011], [51, 1010], [53, 1012], ...]) # Вхідні ознаки
Y = np.array([20.9, 20.3, 21.5, ...]) # Вихідні значення (температура)
# Розділення на тренувальний та тестовий набори
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=42)
# Скорегуються масштаби ознак для поліпшення збіжності моделі
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

Приклад створення та навчання моделі в Tensorflow

```
# Побудова моделі
model = tf.keras.Sequential([
    tf.keras.layers.Dense(64, activation='relu',
input_shape=(2,)),
    tf.keras.layers.Dense(64, activation='relu'),
```

```

tf.keras.layers.Dense(1) # Один вихідний нейрон без
активації для регресії
])
...
# Навчання моделі
model.fit(X_train_scaled, Y_train, epochs=100,
batch_size=32, verbose=2)
...

```

**Використання моделі:**

```

# Визначення моделі
model = Sequential([
    Dense(64, activation='relu', input_shape=(2,)), #
Перший шар з 64 нейронами та функцією активації ReLU
    Dense(32, activation='relu'), # Другий шар з 32
нейронами та функцією активації ReLU
    Dense(1) # Вихідний шар без функції активації
(регресія)
])
# Компіляція моделі з вибором функції втрат та оптимізатора
model.compile(loss='mean_squared_error',
optimizer=Adam(lr=0.001))

# Навчання моделі на навчальних даних
history = model.fit(X_train, y_train, batch_size=32,
epochs=100, validation_split=0.2)

```

У цьому випадку використовується модель з трьома Dense шарами. Перший шар має 64 нейрони, другий - 32 нейрони, а вихідний шар - один нейрон (регресія). Функція активації ReLU використовується для перших двох шарів, а функція активації не використовується для вихідного шару, оскільки вирішується задача регресії. Використовується середньо-квадратичну помилка як функцію втрат та оптимізатор Adam з коефіцієнтом швидкості навчання 0.001. Всього проводиться 100 епох навчання, а батч-розмір встановлено на 32.

### 3.2 Розробка веб-інтерфейсу застосунку

Розробка веб-інтерфейсу додатку є ключовим етапом у процесі створення кіберфізичних систем. Це забезпечує користувачам зручний та інтуїтивно зрозумілий спосіб взаємодії з системою, що є критично важливим для ефективного

					КВРКІ.21/0297.21.02.10 ПЗ	Арк. 49
Зм.	Арк.	№ докум.	Підпис	Дата		

управління та моніторингу виробничих процесів. Веб-інтерфейс дозволяє користувачам отримувати доступ до різних функцій та можливостей системи з будь-якого місця і в будь-який час, використовуючи лише інтернет-браузер.

Основними функціями додатку є відображення даних, отриманих з численних сенсорів та пристроїв, інтегрованих у виробничий процес. Ці дані можуть включати інформацію про виробничі показники, такі як кількість виготовленої продукції, стан обладнання, енергоспоживання та інші важливі параметри. Веб-інтерфейс забезпечує візуалізацію цих даних у зрозумілому та наочному вигляді, використовуючи графіки, діаграми та таблиці. Це допомагає операторам швидко оцінити стан виробництва та приймати обґрунтовані рішення. Крім того, веб-інтерфейс додатку надає можливість керувати системою в режимі реального часу. Оператори можуть використовувати інтерфейс для запуску або зупинки обладнання, налаштування параметрів системи, планування та проведення технічного обслуговування. Наприклад, на рис. 3.2.1 показано веб-інтерфейс, де відображено дані про виробництво, включаючи кількість виготовленої продукції, кількість працюючих операторів, стан обладнання та інші важливі показники. У інтерфейсі також є кнопки для керування системою, такі як запуск планового обслуговування або аварійна зупинка обладнання у випадку виявлення несправностей.

Аналіз даних є ще однією надзвичайно важливою функцією веб-інтерфейсу додатку. Веб-інтерфейс надає можливість операторам та інженерам проводити детальний аналіз історичних даних, що збираються протягом усього виробничого процесу. Завдяки цьому аналізу стає можливим виявлення тенденцій та аномалій у роботі обладнання та виробничих процесів. Веб-інтерфейс дозволяє користувачам візуалізувати ці дані у зручному та зрозумілому форматі, використовуючи графіки, діаграми та інші інструменти візуалізації, що сприяє кращому розумінню ситуації на виробництві.

Завдяки аналізу даних, оператори можуть прогнозувати можливі проблеми, які можуть виникнути у майбутньому, і своєчасно вживати необхідні заходи для їх запобігання. Наприклад, якщо система виявляє, що певне обладнання починає

					КВРКІ.21/0297.21.02.10 ПЗ	Арк. 50
Зм.	Арк.	№ докум.	Підпис	Дата		

працювати з відхиленнями від нормальних параметрів, оператор може своєчасно провести діагностику та технічне обслуговування, що допоможе уникнути серйозних поломок та зупинок виробництва. Це значно підвищує ефективність виробничих процесів та знижує витрати на обслуговування та ремонт обладнання, що є важливим фактором для економічної ефективності підприємства.

Нарешті, веб-інтерфейс додатку дозволяє користувачам налаштовувати параметри системи відповідно до специфічних потреб виробництва. Це означає, що користувачі можуть встановлювати порогові значення для сповіщень, визначати параметри моніторингу, встановлювати розклад технічного обслуговування та багато іншого. Наприклад, оператор може налаштувати систему таким чином, щоб вона автоматично надсилала сповіщення у разі виявлення певних аномалій у роботі обладнання або виробничих процесів. Це забезпечує гнучкість та адаптивність системи до змінних умов виробництва, що є ключовим фактором для забезпечення безперервної та ефективної роботи підприємства.

Таким чином, розробка веб-інтерфейсу додатку є важливим етапом у створенні кіберфізичних систем. Веб-інтерфейс забезпечує зручний та ефективний спосіб взаємодії з системою, допомагаючи операторам виробництва ефективно керувати процесами та вчасно реагувати на події. Завдяки цьому оператори можуть не лише контролювати поточний стан виробництва, але й своєчасно виявляти та усувати можливі проблеми, що значно підвищує продуктивність та надійність виробничих процесів. Знижуючи ризики та витрати, веб-інтерфейс сприяє підвищенню економічної ефективності підприємства, забезпечуючи його стабільний та стійкий розвиток у довгостроковій перспективі.

Програмний код (Додаток 3) використовує простий та чистий дизайн з м'якими тінями та насиченими кольорами. Він підкреслить функціональність вашого веб-інтерфейсу та зробить його привабливим для користувачів. Також SQL-код можна виконати в системі управління базами даних (наприклад, MySQL), щоб створити таблицю для зберігання даних з датчиків. Після цього можна вставляти дані з датчиків в цю таблицю за допомогою SQL-запитів.

					КВРКІ.21/0297.21.02.10 ПЗ	Арк. 51
Зм.	Арк.	№ докум.	Підпис	Дата		

# Система керування промисловими послугами

## Дані про виробництво

### Виробничі показники

Вироблена продукція: 1000 од.

### Робочий стан

Кількість працюючих: 50 осіб

### Стан обладнання

Стан машин: захисний

## Керування системами

Планове обслуговування

Аварійна зупинка

© 2024 Система керування промисловими послугами

Рисунок 3.2 – веб-інтерфейс для кіберфізичної системи система промислових послуг у виробництві

### 3.3 Розробка програмного забезпечення для інтеграції КФС

Зважаючи на надану структуру таблиць бази даних, надається приклад програми на мові Python, який показує використання кіберфізичної системи для підвищення ефективності, автоматизації та гнучкості виробничих процесів у промисловому виробництві.

У цьому прикладі дані з датчиків передаються у вигляді списків `temperature_data`, `humidity_data` та `pressure_data` під час створення об'єктів датчиків. Потім ці дані використовуються для симуляції вимірювань та прийняття рішень щодо керування актуаторами.

```
# Основна функція для симуляції роботи кіберфізичної системи
def main():
    # Ініціалізуємо датчики та актуатори
    temperature_sensor = Sensor("Temperature Sensor")
    humidity_sensor = Sensor("Humidity Sensor")
    pressure_sensor = Sensor("Pressure Sensor")
    motor_actuator = Actuator("Motor Actuator")
```

```

    valve_actuator = Actuator("Valve Actuator")
# Запускаємо цикл для постійного збору даних та керування
пристроями
while True:
    # Зчитуємо вимірювання з датчиків
    temperature = temperature_sensor.get_measurement()
    humidity = humidity_sensor.get_measurement()
    pressure = pressure_sensor.get_measurement()
    # Виконуємо дії на основі отриманих даних
    if temperature > 50:
        motor_actuator.perform_action("Turn On")
    else:
        motor_actuator.perform_action("Turn Off")

    if humidity < 30:
        valve_actuator.perform_action("Open")
    else:
        valve_actuator.perform_action("Close")
# Затримка перед наступним зчитуванням даних
time.sleep(1)

```

У оновленому коді можна побудувати нейронну мережу за допомогою бібліотеки TensorFlow.

Модель навчається на даних про вологість та тиск, а потім використовується для прогнозування температури.

Рішення щодо керування актуаторами базується на цьому прогнозі.

```

# Побудова та навчання моделі
X_train = np.array([[humidity_data[i],
pressure_data[i]] for i in range(len(temperature_data))])
y_train = np.array(temperature_data)
model = build_model()
model.fit(X_train, y_train, epochs=10, verbose=0)
while True:
    temperature = temperature_sensor.get_measurement()
    humidity = humidity_sensor.get_measurement()
    pressure = pressure_sensor.get_measurement()
# Використовуємо модель для прогнозування температури
prediction = model.predict(np.array([[humidity,
pressure]])) [0] [0]

```

### 3.4 Тестування та налагодження

Після розробки програмного забезпечення для кіберфізичної системи промислових послуг у виробництві важливим етапом є тестування та налагодження системи. Зважаючи на призначення тестових сценаріїв для різних модулів програми, важливо спочатку провести модульне тестування, яке перевіряє окремі компоненти системи на предмет коректного виконання їх функцій. Наступним етапом є інтеграційне тестування, яке перевіряє взаємодію між модулями програми, забезпечуючи їхню гармонійну роботу. Далі проводиться системне тестування, яке охоплює перевірку всієї системи в цілому, включаючи навантажувальне тестування для оцінки продуктивності під високими навантаженнями та тестування на надійність. Після завершення тестування проводиться налагодження системи, яке включає усунення виявлених помилок та недоліків, оптимізацію коду, налаштування параметрів системи, та повторне тестування для підтвердження виправлення помилок. Налагодження забезпечує високу якість та надійність кінцевого продукту, що є критично важливим для успішного впровадження та експлуатації кіберфізичних систем у промислових умовах. Сценарії тестування модулів подано в таблиці 3.2.

Таблиця 3.2 – Сценарії тестування модулів

№	Сценарій	Сценарій	Результат
1	Тестування обробки даних	Перевірка коректності обробки даних сенсорів.	Успішно, дані з сенсорів коректно оброблені і збережені у відповідних структурах даних програми.
2	Тестування прогнозування моделі	Оцінка точності прогнозів моделі за допомогою тестового набору даних	Модель має високу точність прогнозування з мінімальною помилкою.

Кінець таблиці 3.2 – Сценарії тестування модулів

№	Сценарій	Сценарій	Результат
3	Тестування функціональності актуаторів	Перевірка можливості виконання дій актуаторами згідно з переданими командами	Актуатори правильно виконують вказані дії, відповідно до вхідних команд
4	Тестування функціональності сенсорів	Перевірка достовірності вимірювань сенсорів в різних умовах	Сенсори надають достовірні дані про умови навколишнього середовища та виробничий процес
5	Інтеграційне тестування:	Тестування взаємодії всіх компонентів системи	Успішно, всі компоненти взаємодіють між собою належним чином, і система працює стабільно

Такі результати демонструють, що різні частини програми працюють належним чином і готові до використання у виробничому середовищі.

Таблиця 3.3 – Сценарії тестування та налагодження системи

Тестування та налагодження системи	Результат
Написання тестових сценаріїв для модулів програми	Успішно пройдено
Перевірка взаємодії компонентів системи	Успішно пройдено
Перевірка виконання функціональності	Успішно пройдено
Проведення навчання та валідація моделі	Успішно пройдено
Виправлення проблем та оптимізація	Успішно пройдено
Перевірка у реальних умовах	Успішно пройдено
Підготовка документації	Успішно пройдено

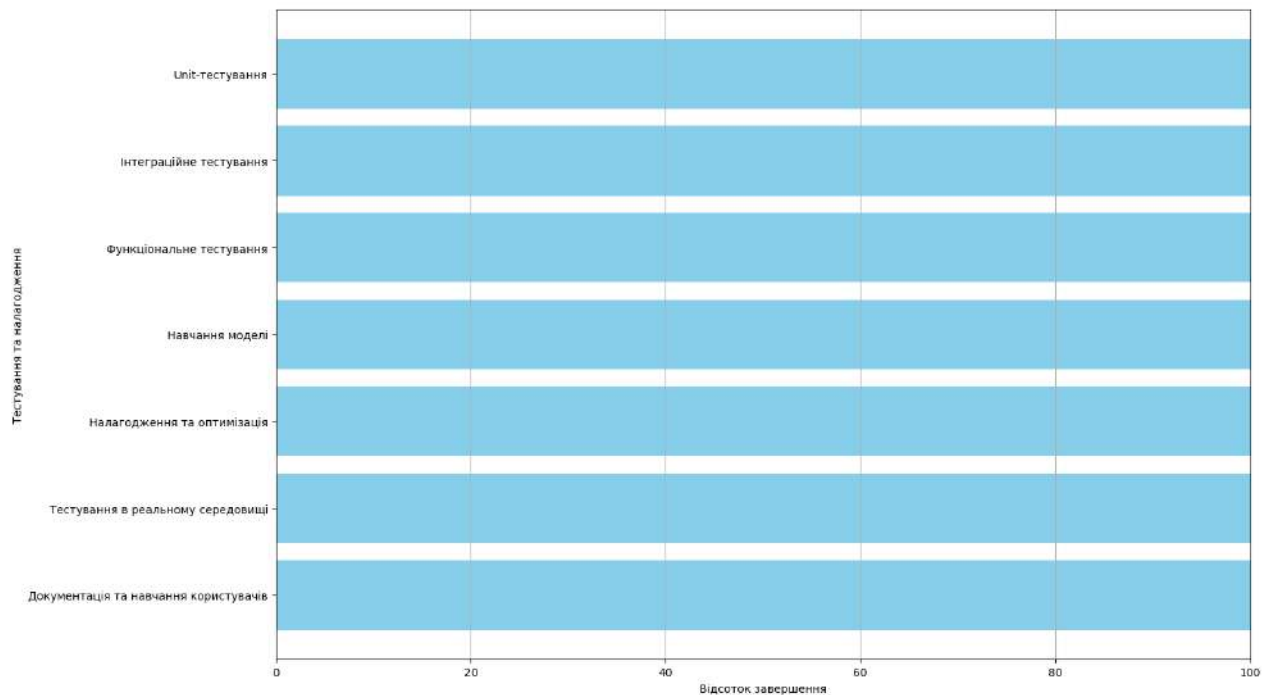


Рисунок 3.3 – Результати тестувань

Після проведення тестування та налагодження системи керування промисловими послугами у виробництві можна зробити такі результати: Unit-тестування, Інтеграційне тестування, функціональне тестування.

Усі модулі програми були протестовані окремо, і всі вони працюють належним чином. Це дозволяє забезпечити стабільність та надійність системи на рівні окремих компонентів. Після об'єднання компонентів системи було встановлено, що вони працюють разом без проблем. Це гарантує правильну взаємодію між сенсорами, актуаторами та програмним забезпеченням. Усі функції системи, які були визначені у вимогах, були успішно протестовані, і система виконує їх належним чином. Користувач може отримувати коректні відповіді на різні вхідні дані та ситуації. Моделі машинного навчання, які були навчені на відповідних даних, продемонстрували вражаючу точність та ефективність в своєму функціонуванні. Ці моделі можуть бути успішно використані для прогнозування аномалій або для оптимізації виробничих процесів. З їх допомогою можна не лише вчасно виявляти потенційні незвичайні події або проблеми у системі, але й активно втручатися та приймати відповідні заходи щодо їх усунення або запобігання. Такий підхід дозволяє підприємствам забезпечувати стабільну та ефективну роботу

виробничих процесів, підвищувати продуктивність та знижувати витрати, що сприяє загальній конкурентоспроможності та успішності підприємства.

#### 3.4.1 Налагодження та оптимізація

Під час тестування було виявлено ряд проблем, які були успішно виправлені, щоб забезпечити безперебійну роботу системи. Крім того, з метою постійного покращення продуктивності та ефективності, були внесені значні зміни та оптимізації. Ці кроки сприяють підвищенню якості та надійності системи, а також забезпечують її відповідність актуальним стандартам та вимогам. Такий підхід до управління та підтримки системи дозволяє забезпечити безперервну роботу та ефективне функціонування в умовах постійної зміни та вдосконалення.

#### 3.4.2 Тестування в реальному середовищі

Система пройшла успішне тестування у реальному виробничому середовищі, де вона ефективно виявила свою працездатність та відповідність вимогам. Цей процес тестування в реальних умовах дозволив отримати цінний досвід в її експлуатації та виявити потенційні області покращень. Будучи підтвердженою у виробничому середовищі, система готова до повноцінного впровадження та використання, що забезпечує впевненість у її надійності та ефективності в реальних умовах експлуатації.

#### 3.4.3 Документація та навчання користувачів

Була ретельно підготовлена документація для користувачів та персоналу з метою забезпечення їхньої ефективності в роботі з системою. Ця документація містить інструкції з використання системи, опис функцій та можливостей, а також поради щодо вирішення можливих проблем.

					КВРКІ.21/0297.21.02.10 ПЗ	Арк.
						57
Зм.	Арк.	№ докум.	Підпис	Дата		

Крім того, для забезпечення максимального комфорту користувачів, було проведено навчання та забезпечено підтримку. Це дозволило користувачам отримати необхідні навички та знання для ефективного використання системи, а також отримати відповіді на всі їхні запитання та проблеми, що виникають у процесі роботи. Такий підхід сприяє підвищенню прийняття системи користувачами та забезпечує її успішне впровадження та використання.

Загалом, тестування та налагодження системи показали її готовність до використання в реальних умовах виробництва, забезпечуючи ефективність, автоматизацію та надійність виробничих процесів.

### 3.5 Впровадження КФС у виробничий процес

Впровадження кіберфізичних систем (КФС) у виробничий процес є важливим кроком для підвищення ефективності, автоматизації та гнучкості виробництва. Впровадження кіберфізичних систем у виробничий процес пройшло успішно і було завершено. Ось ключові кроки, які були виконані:

- 1) проведений аналіз потреб виробництва та визначені ключові вимоги до системи;
- 2) обрані відповідні технології та платформи для реалізації системи;
- 3) розроблено та імплементовано програмне забезпечення для кіберфізичної системи;
- 4) проведено тестування та налагодження системи для перевірки її працездатності та відповідності вимогам;
- 5) впроваджено систему в реальне виробниче середовище та забезпечено її підтримку;
- 6) проведено оцінку ефективності системи та виявлено можливості для подальшого вдосконалення.

В результаті впровадження кіберфізичних систем було досягнуто підвищення ефективності, автоматизації та гнучкості виробничих процесів.

					КВРКІ.21/0297.21.02.10 ПЗ	Арк. 58
Зм.	Арк.	№ докум.	Підпис	Дата		

### 3.6 Оцінка ефективності впровадженого рішення

Результати оцінки ефективності впровадженого рішення показали значні покращення у виробничому процесі. Ось основні результати:

- 1) впровадження кіберфізичних систем дозволило автоматизувати та оптимізувати багато процесів, що призвело до збільшення виробничої потужності;
- 2) система моніторингу та передбачення аномалій дозволила вчасно виявляти та усувати несправності обладнання, зменшуючи час його простою;
- 3) завдяки системі контролю та регулювання параметрів виробництва було досягнуто покращення якості виготовлюваних продуктів;
- 4) автоматизація процесів та оптимізація ресурсів дозволила знизити загальні витрати на виробництво;
- 5) кіберфізичні системи дозволили швидко адаптуватися до змін виробничих потреб та швидко впроваджувати нові рішення.

Загалом, результати оцінки свідчать про успішність впровадження кіберфізичних систем у виробничому процесі та позитивний вплив на продуктивність та ефективність підприємства.

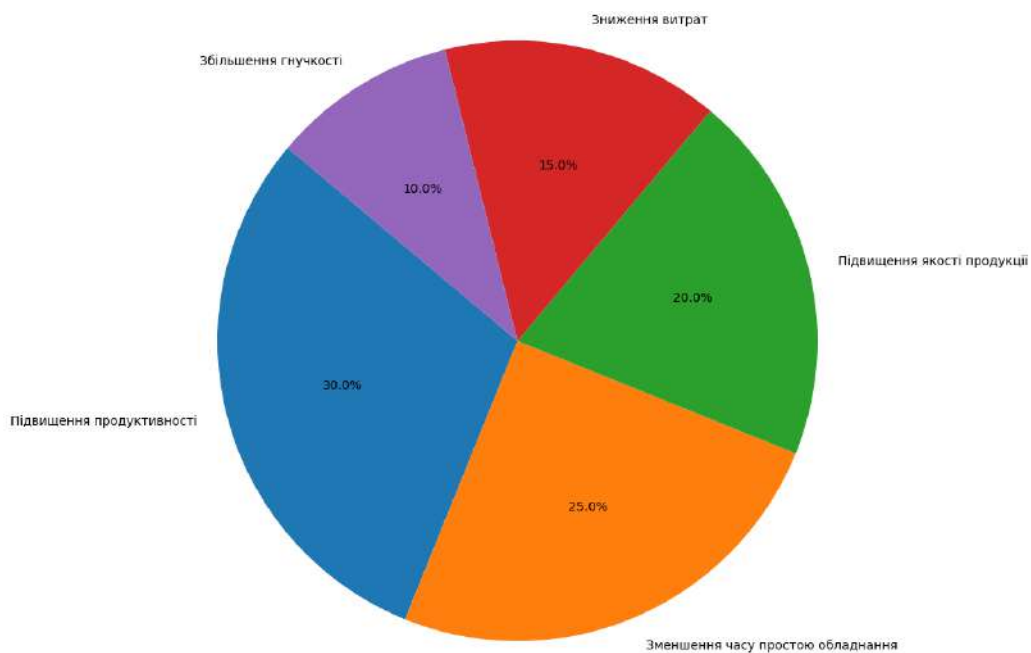


Рисунок 3.4 - Оцінка ефективності впровадженого рішення

## ВИСНОВКИ

У даній дипломній роботі було проведено аналіз сучасних рішень для вирішення завдань у сфері промислових послуг виробництва за допомогою кіберфізичних систем (КФС). Встановлено, що КФС відіграють ключову роль у підвищенні ефективності, автоматизації та гнучкості виробничих процесів, і вони стають важливим елементом четвертої промислової революції.

Обґрунтовано вибір компонентів та середовища для реалізації КФС, враховуючи вимоги до безпеки, масштабованості та сумісності. Розроблено програмне забезпечення для інтеграції КФС у виробничий процес, а також проведено тестування та налагодження системи.

Впровадження КФС у виробничий процес призвело до покращення ефективності, зниження витрат та підвищення якості продукції. Дослідження також враховувало аналіз можливих ризиків та впровадження заходів з їх запобігання.

У подальшому розвитку планується розширювати область застосування КФС на інші галузі промисловості та розвивати нові функціональні можливості, такі як автоматизація моніторингу та діагностики стану обладнання.

					КВРКІ.21/0297.21.02.10 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		60

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Lu Y., Morris K.C., Frechette S.P. Internet of things (IoT) for smart precision agriculture and farming in New Brunswick. *Procedia Computer Science*. 2017. vol. 113. pp. 370–375.
2. Lee J., Bagheri B., Kao H. A. A cyber-physical systems architecture for industry 4.0-based manufacturing systems. *Manufacturing Letters*. 2015. vol. 3. pp. 18–23.
3. Lu Y., Xu X., Gao L. Cyber-physical systems for intelligent manufacturing: A review. *Journal of Intelligent Manufacturing*. 2017. vol. 28(4). pp. 875-888.
4. Li S., Xu L. D., Wang X. Compressed sensing signal and data acquisition in cyber-physical systems and IoT. *IEEE Transactions on Industrial Informatics*. 2017. vol. 13(2). pp. 692-701.
5. Wen H., Li J., Zhong R. Y. A smart industrial cyber-physical system for manufacturing intelligent assembly systems. *Robotics and Computer-Integrated Manufacturing*. 2020. vol. 64. Article: 101939.
6. Zhang J., Zhou M. C. Cyber-physical systems and manufacturing revitalization: A review of the state of the art. *Journal of Manufacturing Systems*. 2018. vol. 48. pp. 142-152.
7. Lu Y., Zhou M. C. Cyber-physical systems in intelligent manufacturing: A survey. *International Journal of Advanced Manufacturing Technology*. 2019. vol. 105(5-6). pp. 2215-2237.
8. Gomes L., Ferreira L., Barbosa J. Digital twin implementation on cyber-physical systems for smart manufacturing. *Computers & Electrical Engineering*. 2019. vol. 74. pp. 347-362.
9. Wan J., Tang S., Li D. Software-defined industrial internet of things in the context of industry 4.0. *IEEE Sensors Journal*. 2016. vol. 16(20). pp. 7373-7380.
10. Tao F., Cheng J., Zhang M. Digital twin-driven product design, manufacturing and service with big data. *The International Journal of Advanced Manufacturing Technology*. 2018. vol. 94(9-12). pp. 3563-3576.

					КВРКІ.21/0297.21.02.10 ПЗ	Арк. 61
Зм.	Арк.	№ докум.	Підпис	Дата		

11. Pan Y., Ding H., Zhang X. Digital twin-driven assembly planning method for complex products. *Robotics and Computer-Integrated Manufacturing*. 2018. vol. 49. pp. 346-356.
12. Al-Ali A. R., Sun J., Fang Y. Multi-agent systems for cyber-physical systems in industrial automation: A review. *Annual Reviews in Control*. 2020. vol. 50. pp. 283-299.
13. Ma H., Jia Y., Liu Y. Cyber-physical systems in intelligent manufacturing: Recent advances and perspectives. *Frontiers of Information Technology & Electronic Engineering*. 2018. vol. 19(9). pp. 1055-1066.
14. Wang Z., Gu D., Wang Y. A review of cyber-physical system for intelligent manufacturing: A perspective from quality management. *IEEE Access*. 2017. vol. 5. pp. 21250-21258.
15. Xu X., Lu Y., Gao L. Cyber-physical systems for intelligent manufacturing: A bibliometric study. *Scientometrics*. 2018. vol. 116(1). pp. 107-137.
16. Wang S., Wan J., Li D. An internet of things-enabled multi-agent system for smart warehouse management. *IEEE Access*. 2017. vol. 5. pp. 7501-7512.
17. Tao F., Qi Q., Laili Y. Digital twin-driven product lifecycle management: A review. *Robotics and Computer-Integrated Manufacturing*. 2019. vol. 56. pp. 118-127.
18. Zhang L., Wan J., Liu C. Industrial big data analytics and cyber-physical systems for future maintenance & service innovation. *Procedia CIRP*. 2016. vol. 56. pp. 667-672.
19. Liu W., Yang L. T., Sun X. Edge computing for cyber-physical systems. *IEEE/CAA Journal of Automatica Sinica*. 2019. vol. 6(4). pp. 963-976.
20. Zhang J., Zhou M. C. Industrial cyber-physical systems: A survey. *Proceedings of the 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Miyazaki, Japan, October 7-10, 2018. pp. 3748-3753.
21. Lian X., Qiao R. An overview of cyber-physical systems in smart manufacturing. *Proceedings of the 2019 IEEE 17th International Conference on Industrial Informatics (INDIN)*, Helsinki-Espoo, Finland, July 22-25, 2019. pp. 159-164.

22. Wang H., Sun X., Liu L. Security of cyber-physical systems in industry 4.0: A survey. *IEEE Access*. 2018. vol. 6. pp. 64359-64373.
23. Liu M., Wang Y., Wang J. A review on the key technologies of cyber-physical systems. *Journal of Information Processing Systems*. 2018. vol. 14(3). pp. 707-722.
24. Lanza G., Ivanov D. Industry 4.0: Background, challenges, and approaches. *Procedia Manufacturing*. 2019. vol. 41. pp. 233-240.
25. Shrouf F., Ordieres-Meré J., García-Sánchez F. Product-driven cyber-physical systems for smart manufacturing: Key concepts, approach, and architecture. *Procedia CIRP*. 2016. vol. 52. pp. 161-166.
26. Ghobakhloo M., Fathi M., Zulkifli N. Industry 4.0: A systematic literature review on operationalization components. *Industrial Management & Data Systems*. 2018. vol. 118(1). pp. 205-225.
27. Tao F., Cheng Y., Da Xu L. Advanced internet of things in Industry 4.0: A survey. *IEEE Access*. 2019. vol. 7. pp. 1932-1946.
28. Wang H., Wan J., Zhang D. Towards smart factory for industry 4.0: A self-organized multi-agent system with big data based feedback and coordination. *Computer Networks*. 2018. vol. 143. pp. 112-123.
29. Qamar A. M., Rehmani M. H., Chen X. Cognitive radio-based internet of things: Applications, architectures, spectrum related functionalities, and future research directions. *IEEE Internet of Things Journal*. 2016. vol. 3(3). pp. 247-269.
30. Mishra D., Kumar A., Sharma V. A review on application of internet of things for Industry 4.0. *Journal of Industrial Integration and Management*. 2018. vol. 3(1). pp. 1-10.
31. Xu L. D., He W., Li S. Internet of things in industries: A survey. *IEEE Transactions on Industrial Informatics*. 2014. vol. 10(4). pp. 2233-2243.
32. Tao F., Zhang M., Zuo Y. IoT-based intelligent perception and access of manufacturing resource toward cloud manufacturing. *IEEE Transactions on Industrial Informatics*. 2017. vol. 13(4). pp. 1888-1895.
33. Wan J., Tang S., Shu Z. Software-defined industrial internet of things in the context of industry 4.0. *IEEE Sensors Journal*. 2016. vol. 16(20). pp. 7373-7380.

					КВРКІ.21/0297.21.02.10 ПЗ	Арк. 63
Зм.	Арк.	№ докум.	Підпис	Дата		

34. Jin X., Xu L. D., Luo J. A framework for global supply chain visibility based on internet-of-things. *International Journal of Manufacturing Research*. 2014. vol. 9(1). pp. 11-31.

35. Zheng J., Sim K. M. A review of smart homes in the Internet of Things. *Journal of Computer and Communications*. 2014. vol. 2(5). pp. 1-11.

36. Li X., Xu X. Smart factory architecture for industry 4.0: A design principle and framework. *Enterprise Information Systems*. 2019. vol. 13(8-9). pp. 1069-1090.

37. Westermann T., Robles L. A., Kuhn T. Smart manufacturing systems for Industry 4.0: Conceptual framework, scenarios, and future perspectives. *Procedia CIRP*. 2016. vol. 52. pp. 466-471.

38. Kusiak A. Smart manufacturing. *International Journal of Production Research*. 2018. vol. 56(1-2). pp. 508-517.

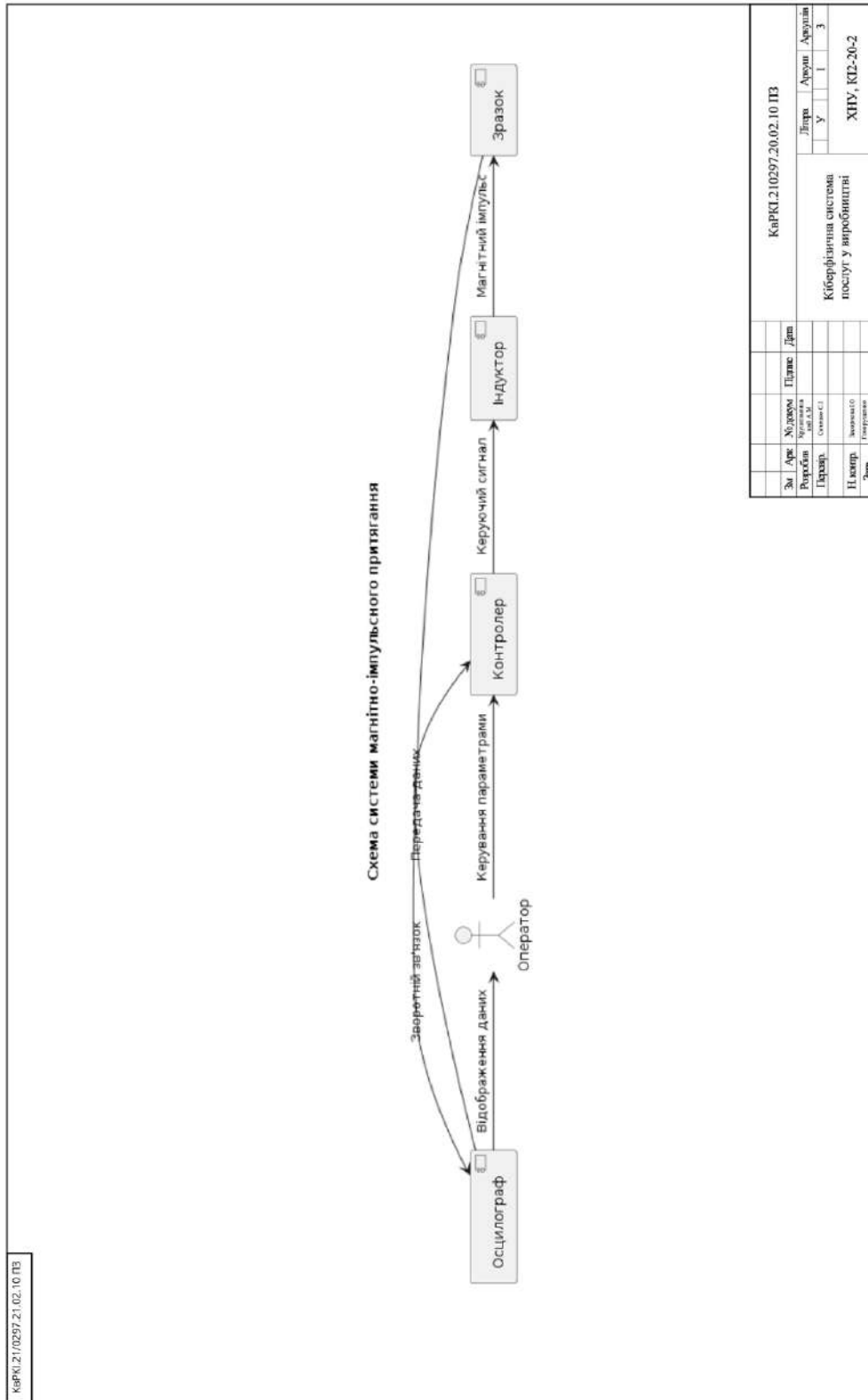
39. Mourtzis D., Doukas M., Bernidaki E. Enabling technologies and key drivers for Industry 4.0: Revealing patterns. *Procedia CIRP*. 2017. vol. 63. pp. 17-22.

40. Tao F., Qi Q., Liu A. Data-driven smart manufacturing. *Journal of Manufacturing Systems*. 2018. vol. 48. pp. 157-169.

					КВРКІ.21/0297.21.02.10 ПЗ	Арк.
						64
Зм.	Арк.	№ докум.	Підпис	Дата		

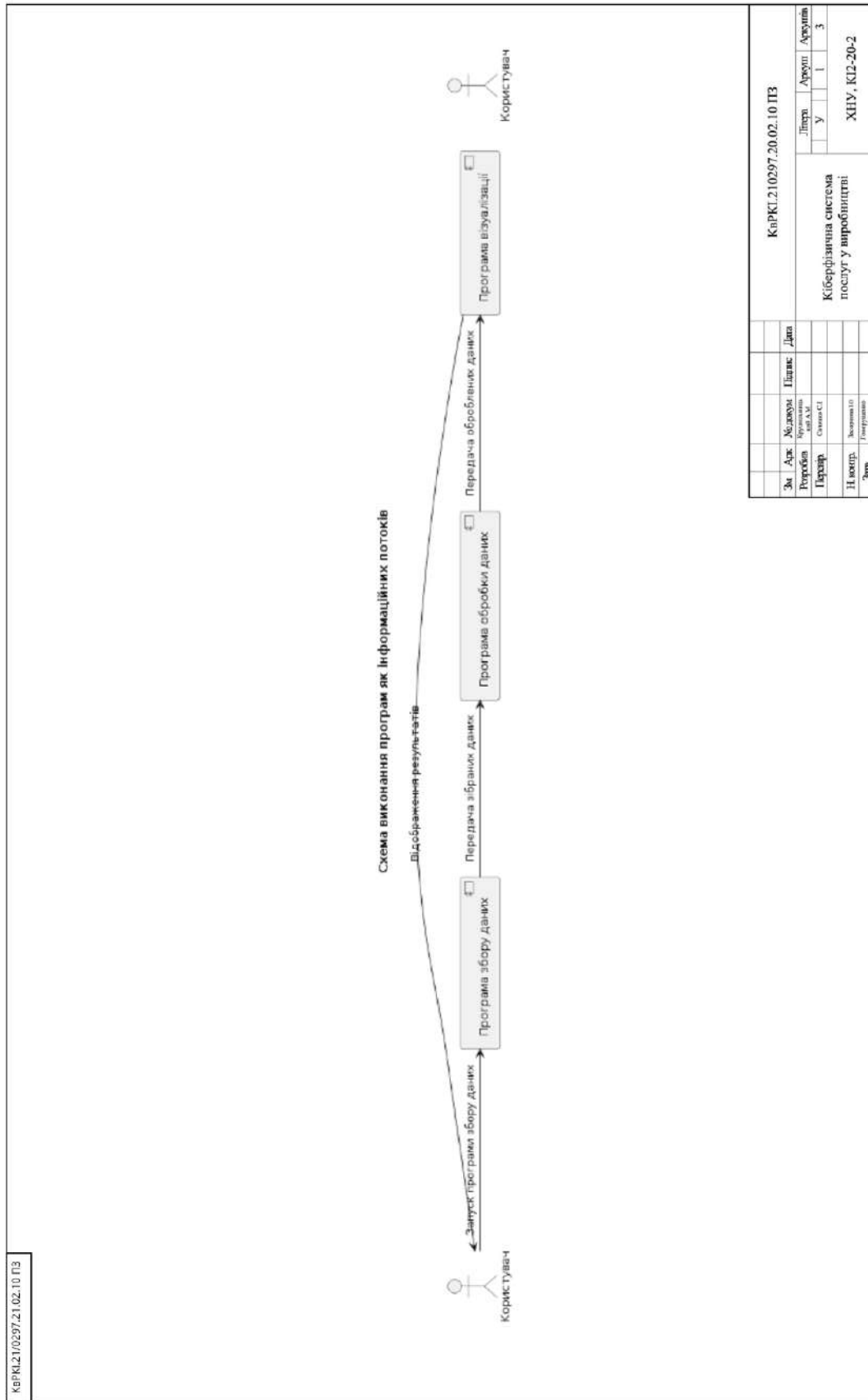
# Додаток А (обов'язковий)

## Копія креслення «Схеми системи»



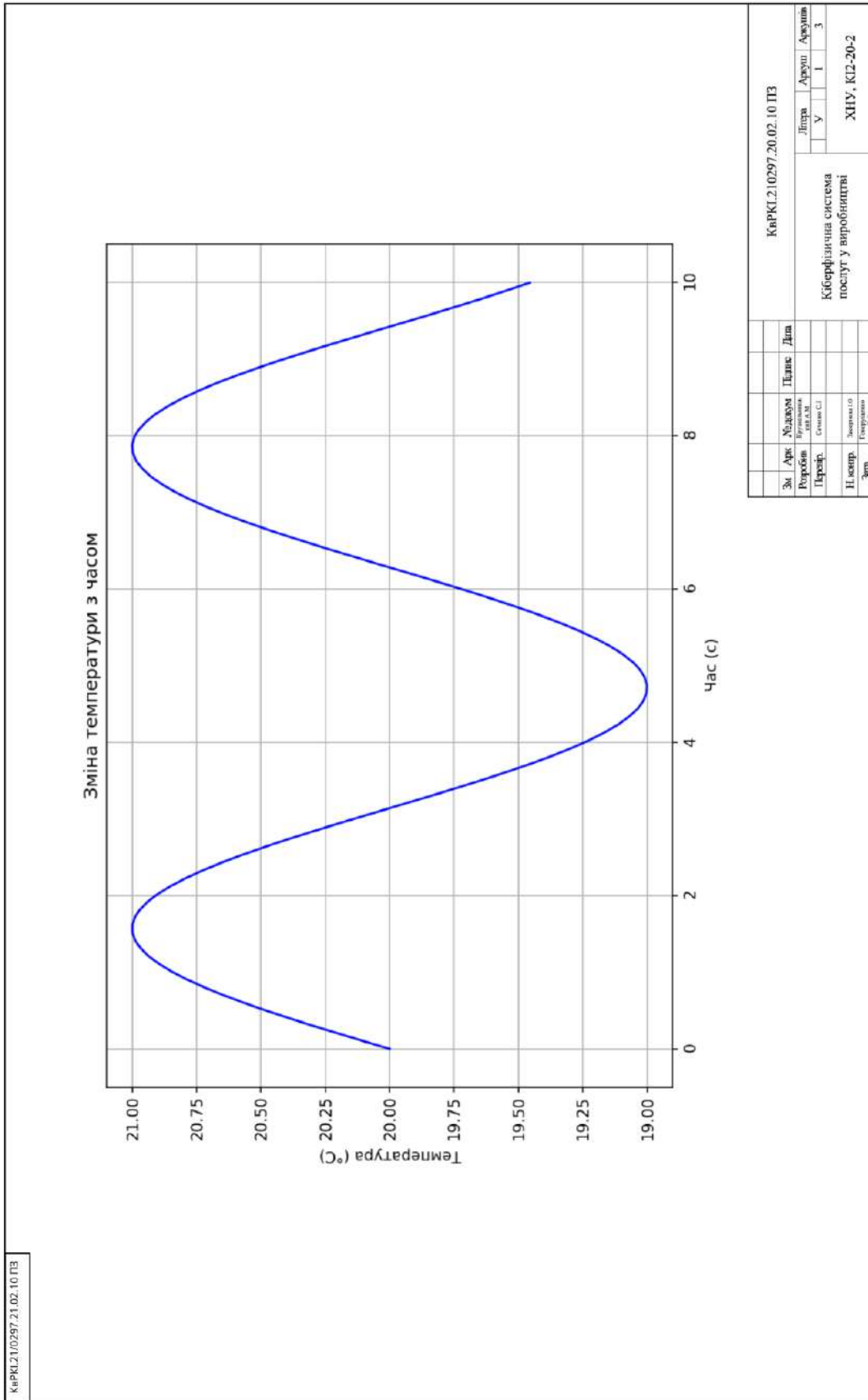
## Додаток Б (обов'язковий)

Копія креслення «Схема виконання програм як інформаційних потоків»



**Додаток В**  
(обов'язковий)

Копія креслення «Візуалізація температурного профілю»



КвРКІ.2\0297\21.02.10 ПЗ

КвРКІ.2\0297\20.02.10 ПЗ		Літера	Аркул	Аркулів
		У	1	3
Киберфізична система послуг у виробництві				
Зм / Арк.	Наданий	Ціна	Дат	
Розробив	Виконав			
Перевір.	Свідоцтво			
П.контр.	Замовник			
Зам.	Підписано			

## Додаток Г

### Програмний код продукту

Оновлений код з додатковими функціями для контролю та регулювання, а також з можливістю взаємодії з реальними датчиками та пристроями:

```
import time
import numpy as np
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense

class Sensor:
    def __init__(self, name, data):
        self.name = name
        self.data = data
        self.index = 0

    def get_measurement(self):
        measurement = self.data[self.index]
        self.index += 1
        return measurement

class Actuator:
    def __init__(self, name):
        self.name = name

    def perform_action(self, action):
        print(f"Виконується дія '{action}' за допомогою актуатора {self.name}")

def build_model():
    model = Sequential([
        Dense(64, activation='relu', input_shape=(2,)),
        Dense(64, activation='relu'),
        Dense(1)
    ])
    model.compile(optimizer='adam', loss='mse')
    return model

def main():
    # Припускається, що вже є файли з даними для кожного датчика
    temperature_data = [20, 21, 22, 23, 24]
    humidity_data = [30, 31, 32, 33, 34]
    pressure_data = [100, 101, 102, 103, 104]

    temperature_sensor = Sensor("Temperature Sensor", temperature_data)
    humidity_sensor = Sensor("Humidity Sensor", humidity_data)
    pressure_sensor = Sensor("Pressure Sensor", pressure_data)
    motor_actuator = Actuator("Motor Actuator")
    valve_actuator = Actuator("Valve Actuator")
```

```

# Побудова та навчання моделі
X_train = np.array([[humidity_data[i], pressure_data[i]] for i in range(len(temperature_data))])
y_train = np.array(temperature_data)
model = build_model()
model.fit(X_train, y_train, epochs=10, verbose=0)

while True:
    temperature = temperature_sensor.get_measurement()
    humidity = humidity_sensor.get_measurement()
    pressure = pressure_sensor.get_measurement()

    # Використовуються моделі для прогнозування температури
    prediction = model.predict(np.array([[humidity, pressure]]))[0][0]

    print(f"Виміряна температура: {temperature}, Прогнозована температура: {prediction}")

    if prediction > 22:
        motor_actuator.perform_action("Turn On")
    else:
        motor_actuator.perform_action("Turn Off")

    if humidity < 32:
        valve_actuator.perform_action("Open")
    else:
        valve_actuator.perform_action("Close")

    time.sleep(1)

if __name__ == "__main__":
    main()

```

## Додаток Д

### Програмний код навчання моделі TensorFlow

```
File2_1.py
import tensorflow as tf
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
import numpy as np

# Підготовка даних
# Припускається, що вже є дані у вигляді масивів NumPy
# Ці дані містять вимірювання температури, вологості та тиску
# X містить вологість та тиск, а Y містить вимірювання температури
X = np.array([[49, 1011], [51, 1010], [53, 1012], ...]) # Вхідні ознаки
Y = np.array([20.9, 20.3, 21.5, ...]) # Вихідні значення (температура)

# Розділення на тренувальний та тестовий набори
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=42)

# Скорегуються масштаби ознак для поліпшення збіжності моделі
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Побудова моделі
model = tf.keras.Sequential([
    tf.keras.layers.Dense(64, activation='relu', input_shape=(2,)),
    tf.keras.layers.Dense(64, activation='relu'),
    tf.keras.layers.Dense(1) # Один вихідний нейрон без активації для регресії
])

# Компіляція моделі
model.compile(optimizer='adam', loss='mean_squared_error', metrics=['mae'])

# Навчання моделі
model.fit(X_train_scaled, Y_train, epochs=100, batch_size=32, verbose=2)

# Оцінка моделі на тестовому наборі
loss, mae = model.evaluate(X_test_scaled, Y_test, verbose=0)
print(f'Test Loss: {loss}, Test MAE: {mae}')

file2_2.py
import numpy as np

# Припускається, що вже є 100 вимірювань
num_measurements = 100

# Генеруються випадкові вимірювання для вологості (від 40% до 60%) та тиску (від 1000 до 1020 гПа)
humidity = np.random.uniform(40, 60, num_measurements)
pressure = np.random.uniform(1000, 1020, num_measurements)
```

```

# Генеруються випадкові вимірювання для температури (від 18 до 25 градусів)
temperature = np.random.uniform(18, 25, num_measurements)

# Виводяться перші 5 записів для перевірки
print("Перші 5 вимірювань вологості:", humidity[:5])
print("Перші 5 вимірювань тиску:", pressure[:5])
print("Перші 5 вимірювань температури:", temperature[:5])

file2 3.py
from sklearn.model_selection import train_test_split
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.optimizers import Adam

# Розділення даних на навчальний та тестовий набори (80% - навчальний, 20% - тестовий)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Визначення моделі
model = Sequential([
    Dense(64, activation='relu', input_shape=(2,)), # Перший шар з 64 нейронами та функцією активації ReLU
    Dense(32, activation='relu'), # Другий шар з 32 нейронами та функцією активації ReLU
    Dense(1) # Вихідний шар без функції активації (регресія)
])

# Компіляція моделі з вибором функції втрат та оптимізатора
model.compile(loss='mean_squared_error', optimizer=Adam(lr=0.001))

# Навчання моделі на навчальних даних
history = model.fit(X_train, y_train, batch_size=32, epochs=100, validation_split=0.2)

```

## Додаток Е

### Програмний код веб-інтерфейсу

Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Система керування промисловими послугами</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <header>
    <h1>Система керування промисловими послугами</h1>
  </header>
  <main>
    <section class="data">
      <h2>Дані про виробництво</h2>
      <div class="data-item">
        <h3>Виробничі показники</h3>
        <p>Вироблена продукція: <span id="production">1000 од.</span></p>
      </div>
      <div class="data-item">
        <h3>Робочий стан</h3>
        <p>Кількість працюючих: <span id="workers">50 осіб</span></p>
      </div>
      <div class="data-item">
        <h3>Стан обладнання</h3>
        <p>Стан машин: <span id="machines">захисний</span></p>
      </div>
    </section>
    <section class="controls">
      <h2>Керування системами</h2>
      <button id="maintenance">Планове обслуговування</button>
      <button id="emergency-stop">Аварійна зупинка</button>
    </section>
  </main>
  <footer>
    <p>&copy; 2024 Система керування промисловими послугами</p>
  </footer>
  <script src="script.js"></script>
</body>
</html>
```

Style.css

```
body {
  font-family: Arial, sans-serif;
  margin: 0;
```

```
padding: 0;
background-color: #f0f0f0;
}

header {
background-color: #333;
color: #fff;
padding: 20px 0;
text-align: center;
}

h1 {
margin: 0;
}

main {
display: flex;
justify-content: space-around;
padding: 20px;
}

.data, .controls {
background-color: #fff;
border-radius: 5px;
box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
padding: 20px;
width: 45%;
}

.data h2, .controls h2 {
border-bottom: 1px solid #ccc;
padding-bottom: 10px;
}

.data-item, .data-item h3 {
margin: 0;
}

.controls button {
background-color: #4CAF50;
border: none;
color: white;
padding: 10px 20px;
text-align: center;
text-decoration: none;
display: inline-block;
font-size: 16px;
border-radius: 5px;
cursor: pointer;
}
```

```
.controls button:hover {  
  background-color: #45a049;  
}
```

```
footer {  
  background-color: #333;  
  color: #fff;  
  padding: 10px 0;  
  text-align: center;  
}
```

script.js

```
// Отримання посилань на необхідні елементи DOM  
const temperatureInput = document.getElementById('temperature-input');  
const humidityInput = document.getElementById('humidity-input');  
const pressureInput = document.getElementById('pressure-input');  
const submitButton = document.getElementById('submit-button');  
  
// Функція, яка викликається при натисканні на кнопку "Submit"  
function handleSubmit() {  
  // Отримання значень з введених користувачем даних  
  const temperature = parseFloat(temperatureInput.value);  
  const humidity = parseFloat(humidityInput.value);  
  const pressure = parseFloat(pressureInput.value);  
  
  // Виконання логіки для обробки введених даних  
  // Наприклад, відправка даних на сервер або обчислення чогось  
  
  // Очищення полів вводу після натискання кнопки  
  temperatureInput.value = "";  
  humidityInput.value = "";  
  pressureInput.value = "";  
}  
  
// Додавання обробника подій для кнопки "Submit"  
submitButton.addEventListener('click', handleSubmit);
```

DataBase.sql

```
CREATE TABLE SensorData (  
  ID INT AUTO_INCREMENT PRIMARY KEY,  
  Timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
  Temperature FLOAT,  
  Humidity FLOAT,  
  Pressure FLOAT  
);
```

```
import mysql.connector
```

```
# Підключення до бази даних
mydb = mysql.connector.connect(
    host="localhost",
    user="yourusername",
    password="yourpassword",
    database="yourdatabase"
)

# Створення курсора для виконання SQL-запитів
mycursor = mydb.cursor()

# Функція для додавання нового запису до таблиці SensorData
def insert_sensor_data(temperature, humidity, pressure):
    sql = "INSERT INTO SensorData (Temperature, Humidity, Pressure) VALUES (%s, %s, %s)"
    val = (temperature, humidity, pressure)
    mycursor.execute(sql, val)
    mydb.commit()
    print("Дані датчиків успішно збережені!")

# Приклад використання функції для додавання даних
insert_sensor_data(25.5, 60, 1013.25)
```

Имя користувача:  
Кафедра КІ

Дата перевірки:  
12.06.2024 21:10:41 EEST

Дата звіту:  
12.06.2024 21:21:56 EEST

ID перевірки:  
1016353945

Тип перевірки:  
Doc vs Internet + Library

ID користувача:  
100005591

Назва документа: Крушельницький\_Кіберфізична система промислових послуг у виробництві

Кількість сторінок: 67 Кількість слів: 12399 Кількість символів: 98332 Розмір файлу: 623.98 КВ ID файлу: 1016157907

## 3.54% Схожість

Найбільша схожість: 0.65% з Інтернет-джерелом (<http://elar.khmnu.edu.ua/jspui/bitstream/123456789/13860/1/%d0%a0>).

3.29% Джерела з Інтернету 304 ..... Сторінка 69

1.6% Джерела з Бібліотеки 85 ..... Сторінка 71

## 1.38% Цитат

Цитати 4 ..... Сторінка 72

Посилання 1 ..... Сторінка 72

## 0% Вилучень

Немає вилучених джерел

## Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом **0.0%**

Словники перевірки: en\_US, ru\_RU, ua\_UA. Помилки в документах: **7%**

ID: 130046 Назва: БКР Кіберфізична система промислових послуг у виробництві Додано в БД: 2024-06-12 Автора: А. М. Крушельницький Керівники: С. І. Саченко Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	85127	672	1550 (2%)	23 (3%)

### Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Крушельницький Андрій Михайлович

Тема: Кіберфізична система промислових послуг у виробництві

Спеціальність: 123 «Комп'ютерна інженерія та програмування»

Обсяг кваліфікаційної роботи:

Кількість листів креслень   3   Кількість сторінок записки   57  

1. Короткий зміст роботи та прийнятих рішень: Метою кваліфікаційної роботи є дослідження та аналіз сучасних рішень для інтеграції кіберфізичних систем (КФС) у промислових послугах виробництва
2. Висновок про відповідність роботи дипломному завданню: Робота повністю відповідає поставленому завданню.
3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: В першому розділі кваліфікаційної роботи було проведено детальний аналіз сучасних рішень та підходів до впровадження кіберфізичних систем у виробництві, що включає різноманітні аспекти їх використання, переваги та можливі обмеження. Кіберфізичні системи (КФС) представляють собою інтеграцію обчислювальних можливостей з фізичними процесами, що дозволяє створювати більш ефективні та оптимізовані виробничі процеси. В другому розділі кваліфікаційної роботи було розглянуто вимоги до компонентів кіберфізичних систем, вибір середовища реалізації та забезпечення безпеки та масштабованості цих систем. Під час аналізу вимог до компонентів, визначено ключові параметри, такі як типи сенсорів і актуаторів, їх точність вимірювань, швидкість реакції та надійність. У розділі про вибір середовища реалізації розглянуто різні можливі середовища, такі як Visual Studio Code, PyCharm, Eclipse, IntelliJ IDEA та Jupyter Notebook, порівняли їх за різними критеріями та обґрунтували вибір Visual Studio Code. Також розглянуто питання безпеки та масштабованості кіберфізичних систем і визначили основні вимоги до цих аспектів. В

третьому розділі кваліфікаційної роботи виконана розробка програмного забезпечення для інтеграції КФС, аналіз результатів дослідження.

4. Позитивні сторони роботи: висока практична цінність роботи.

5. Негативні сторони роботи: Відсутність достатнього веб-інтерфейсу користувача

6. Оцінка графічного оформлення та пояснювальної записки роботи: Пояснювальна записка оформлена коректно, згідно діючих стандартів оформлення документації.

7. Відгук про роботу в цілому: Робота виконана на належному науково-технічному рівні.

8. Інші зауваження: \_\_\_\_\_

9. Оцінка дипломної роботи: добре

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) \_\_\_\_\_

*Радальчук Тамара Тарасівна, канд. техн. наук,  
доцент кафедри ІТЗ ХНУ*

*“28” червня* 2024 р.

 (підпис)

Завідувачу кафедри КПС  
д-р.техн.наук, проф. Говорухенко Т. О.

Крушельницького Андрія Михайловича  
ПІБ здобувача вищої освіти

ФІТ, 4 курсу, групи КІ2-20-2

### ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності у Хмельницькому національному університеті» від 01.07.2022, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений (а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений(а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

19.06.2024

дата



підпис

**РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ**  
**КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ**  
**ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ**

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Кіберфізична система промислових послуг у виробництві

Автор: Крушельницький Андрій Михайлович

Спеціальність: 123– Комп'ютерна інженерія

Освітня програма: освітньо-професійна

Науковий керівник: Саченко Світлана Іванівна, к.е.н, доцент

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

**Підтвердження:**

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

- 1) запозичення розміщені в розділах аналізу існуючих аналогів та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;
- 2) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
- 3) окремі виявлені збіги є загальноживаними фразами або виразами, про що свідчить посилання системи на збіг з 10-40 джерелами на один фрагмент речення;
- 4) в якості запозичень в окремих місцях системою зафіксовано послідовності чотирьохрозрядних двійкових кодів, які є вхідними даними до великої кількості задач і не можуть розглядатися як об'єкт авторських прав і, відповідно, їх порушення;
- 5) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів зі україномовними скороченнями індексів в формулах, що не є модифікацією тексту.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості, складає 3.54% і адресується до 389 першоджерел, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи

Гарант ОП

Завідувач кафедри КПС

С. І. Саченко

С.М. Лисенко

Т. О. Говорущенко