

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Автоматики і комп'ютеризованих технологій
(повна назва)

Кафедра Комп'ютерно-інтегрованих технологій, автоматизації та робототехніки
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

рівень вищої освіти другий (магістерський)

Розроблення автоматизованої системи управління конвеєрною
лінією із застосуванням засобів людино-машинного інтерфейсу з
урахуванням технологічних умов виробництва
(тема)

Виконав:
здобувач 2 року навчання,
групи КІТПВм-24-2

Євген ГЛІБОВ

Спеціальності 174 Автоматизація,
комп'ютерно-інтегровані технології та
робототехніка

Тип програми Освітньо-професійна

Освітня програма Комп'ютерно-
інтегровані технологічні процеси і
виробництва

Керівник доц. Наталія ДЕМСЬКА

Допускається до захисту
Зав. кафедри КІТАР

(підпис)

Ігор НЕВЛЮДОВ

2025р.

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

20 листопада 2025 р.

A handwritten signature in black ink, appearing to read 'Evgen Glebov', written in a cursive style.

Євген ГЛІБОВ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Факультет _____ АКТ _____
Кафедра _____ КІТАР _____
Рівень вищої освіти _____ другий (магістерський) _____
Спеціальність 174 Автоматизація, комп'ютерно-інтегровані технології та
робототехніка _____
Тип програми _____ Освітньо-професійна _____
Освітня програма Комп'ютерно-інтегровані технологічні процеси і
виробництва _____
(шифр і назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри КІТАР _____
(підпис)

« ____ » _____ 2025 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві _____ Глебову Євгену Олександровичу _____
(прізвище, ім'я, по батькові)

1. Тема роботи Розроблення автоматизованої системи управління
конвеєрною лінією із застосуванням засобів людино-машинного інтерфейсу з
урахуванням технологічних умов виробництва

Затверджена наказом по університету від 10.11.2025 р. № 1029 Ст _____

2. Термін подання здобувачем роботи до екзаменаційної комісії 24.12.2026 р.

3. Вихідні дані до роботи 3.1 Використане програмне забезпечення: мова
програмування JavaScript; бібліотека React.js; Node.js; Electron; 3.2 Для
розробки та тестування моделей буде використовуватися VSCode як основне
середовище для програмування; 3.3 Системні вимоги: оперативна пам'ять
16 ГБ; процесор 4 ядра; вільне місце на диску від 10 ГБ. Дане обладнання
використовується також для забезпечення обчислювальної потужності та
проведення експериментів.

4. Перелік питань, що потрібно опрацювати в роботі _____

4.1 Вступ; 4.2 Аналіз Технічної специфікації та постановка задач реалізації ПЗ
АСУ КЛ; 4.3 Формалізація та попередня підготовка вхідних даних (статичних
та динамічних); 4.4 Обґрунтування архітектури та технологій реалізації ПЗ
АСУ КЛ; 4.5 Реалізація функціонального модуля Обчислення та Управління;
4.6 Розробка структури Архівації та Протоколювання даних;
4.7 Валідація результатів обчислень та аналіз стану конвеєрної лінії; 4.8
Реалізація Людино-машинного Інтерфейсу (НМІ) та Складу вікон;
4.9 Інтеграція модулів Обчислення, Архівації та Формування звітів у
застосунок; 4.10 Тестування та верифікація функціональності ПЗ АСУ КЛ.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій

Графічний матеріал у вигляді презентації – 11 арк. ф. А 4

6. Консультанти розділів роботи


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

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз існуючих технологій та вибір оптимальної технології	01.09.25-23.09.25	виконано
2	Формалізація вхідних/керуючих даних та розробка структури архівування	24.09.25-18.10.25	виконано
3	Реалізація модуля обчислення / керування та контролю вхідних даних	19.10.25-17.11.25	виконано
4	Розробка та інтеграція людино-машинного інтерфейсу у застосунок	17.11.25-22.11.25	виконано
5	Тестування функціональності, оптимізація та підготовка висновків / звітів	23.11.25-30.11.25	виконано
6	Оформлення пояснювальної записки	23.11.25-01.12.25	виконано
7	Подання роботи на нормоконтроль	01.12.25	виконано
8	Подання роботи для перевірки роботи на академічну доброчесність	02.12.25	виконано
9	Подання роботи на рецензію	03.12.25	виконано
10	Подання роботи на підпис зав. кафедри	05.12.25	виконано
11	Подання кваліфікаційної роботи в ЕК		

Дата видачі завдання 01.09.2025 р.

Здобувач


(підпис)

Євген ГЛЄБОВ

Керівник роботи

(підпис)

доц. Наталія ДЕМСЬКА

РЕФЕРАТ

Пояснювальна записка: 99 с., 1 табл., 22 рис., 2 дод., 20 джерел.

АВТОМАТИЗОВАНА СИСТЕМА УПРАВЛІННЯ, КОНВЕЄРНА ЛІНІЯ, НМІ, ТЕХНОЛОГІЧНИЙ МОНІТОРИНГ, АНАЛІЗ ДРЕЙФУ.

Об'єкт дослідження – процес управління технологічними режимами та обладнанням конвеєрної лінії, що включає збір, аналіз даних та формування керуючих впливів.

Предмет дослідження – програмний комплекс АСУ конвеєрною лінією, методи структурування даних, алгоритми виявлення дрейфу параметрів, принципи побудови НМІ та підсистеми протоколювання.

Мета роботи – підвищення надійності та оперативності керування конвеєрною лінією шляхом розроблення спеціалізованого програмного комплексу моніторингу, що забезпечує аналіз дрейфу технологічних параметрів та діагностику стану обладнання.

У першому розділі обґрунтовано використання веб-технологій замість SCADA для створення локального ПЗ.

У другому розділі визначено вимоги до системи та формалізовано задачі виявлення відхилень у роботі обладнання.

У третьому розділі описано реалізацію модулів обчислення, НМІ, архівації та звітності на базі Full-Stack JavaScript.

У четвертому розділі наведено результати тестування валідації даних та розрахунок надійності апаратного забезпечення.

Результат роботи: розроблено та верифіковано АСУ конвеєрною лінією у вигляді автономного додатка.

ABSTRACT

Explanatory Note: 99 p., 1 tabl., 22 fig., 2 appendices, 20 sources.

AUTOMATED CONTROL SYSTEM, CONVEYOR LINE, HMI, TECHNOLOGICAL MONITORING, DRIFT ANALYSIS.

The object of study is the process of managing technological modes and equipment status of a conveyor line, including data collection, analysis, and the generation of control actions.

The subject of study is the software complex of the Conveyor Line ACS, data structuring methods, algorithms for parameter drift detection, HMI design principles, and logging subsystems.

The goal of the work is to improve the reliability and responsiveness of conveyor line control by developing a specialized monitoring software complex that provides analysis of technological parameter drift and equipment diagnostics.

The first chapter justifies the use of modern web technologies instead of classical SCADA for creating local software.

The second chapter defines the system requirements and formalizes the tasks for detecting hidden deviations in equipment operation.

The third chapter describes the software implementation of calculation, HMI, archiving, and reporting modules based on the Full-Stack JavaScript stack.

The fourth chapter presents the results of data validation testing and the calculation of hardware reliability indicators.

Result of the work: an automated conveyor line control system has been developed and verified in the form of a standalone software application.

ЗМІСТ

Перелік скорочень	8
Вступ.....	10
1 Аналіз існуючих підходів до автоматизації конвеєрних ліній.....	13
1.1 Використання промислових контролерів (PLC) та SCADA-систем....	13
1.2 Обґрунтування вибору комунікаційної платформи TCP/IP	15
1.2.1 Кількісне обґрунтування часу затримки (Latency) НМІ	15
1.2.2 Декомпозиція стеку та гібридний транспортний підхід	16
1.3 Аналіз комерційних рішень та обґрунтування вибору технологічного стеку (Full-Stack JS).....	17
1.3.1 Детальний аналіз існуючих комерційних систем.....	17
1.3.2 Переваги розроблюваної системи на базі Full-Stack JavaScript	18
1.4 Аналіз методів архівації та документування даних в АСУ	19
1.4.1 Вертикальна інтеграція та роль НМІ в архітектурі.....	19
1.4.2 Системи документування, протоколювання та вимоги до кібербезпеки.....	20
1.5 Висновки до розділу	21
2 Аналіз технічного завдання та постановка задач дослідження.....	22
2.1 Аналіз вихідних умов та призначення системи	22
2.2 Формалізація функціональних вимог до ПЗ АСУ КЛ.....	23
2.2.1 Вимоги до модуля збору даних та управління.....	23
2.2.2 Вимоги до модуля аналітичної обробки даних.....	24
2.2.3 Вимоги до системи візуалізації та НМІ	24
2.2.4 Вимоги до підсистеми архівування та звітності	24
2.2.5 Вимоги до адміністративних та сервісних функцій.....	25
2.3 Аналіз нефункціональних вимог та обмежень.....	25
2.4 Постановка науково-дослідних та прикладних задач	26
2.5 Висновки до розділу	27

	6
3 Архітектура та проектування системи керування конвеєрною лінією	28
3.1 Функціональні вимоги до ПЗ АСУ КЛ.....	28
3.1.1 Вимоги до керування та моніторингу в реальному часі	28
3.1.2 Вимоги до архівації даних та звітності.....	28
3.1.3 Вимоги до безпеки та надійності	29
3.2 Архітектура програмного комплексу.....	29
3.2.1 Фізична архітектура системи.....	29
3.2.2 Логічна архітектура застосунку.....	30
3.2.3 Опис компонентів технологічного стеку.....	30
3.3 Проектування структури даних.....	31
3.3.1 Вибір та обґрунтування СКБД.....	31
3.3.2 Опис основних сутностей бази даних.....	31
3.4 Проектування програмних модулів.....	33
3.4.1 Модуль імпорту та валідації даних	33
3.4.2 Модуль розрахунків та аналізу	35
3.4.3 Модуль НМІ та візуалізації.....	38
3.4.4 Модуль аутентифікації та безпеки	38
3.5 Висновки до розділу	39
4 Реалізація програмного забезпечення та дослідження характеристик системи	40
4.1 Реалізація інтерфейсу користувача та основних функцій	40
4.1.1 Автентифікація та головна панель	40
4.1.2 Адміністрування та налаштування системи.....	42
4.1.3 Імпорт та підготовка даних.....	44
4.2 Аналіз даних та формування звітності	45
4.2.1 Аналіз дрейфів та відхилень	45
4.2.2 Генерація звітної документації.....	47
4.3 Розрахунок надійності апаратного забезпечення системи	49
4.3.1 Вихідні дані та методика розрахунку	49
4.3.2 Розрахунок надійності одиночної робочої станції	50

	7
4.3.3 Розрахунок коефіцієнта готовності та вплив ремонту.....	51
4.3.4 Розрахунок надійності системи з гарячим резервуванням.....	51
4.3.5 Вплив планового технічного обслуговування.....	52
4.4 Охорона праці та ергономічне забезпечення роботи оператора.....	53
4.5 Висновки до розділу.....	54
Висновки.....	56
Перелік джерел посилання.....	58
Додаток А Лістинг програми.....	61
Додаток Б Апробація наукових результатів дослідження.....	91
Додаток В Демонстраційний матеріал.....	97

ПЕРЕЛІК СКОРОЧЕНЬ

АРМ – автоматизоване робоче місце;
АСУ – автоматизована система управління;
БД – база даних;
ІТ – інформаційні технології;
КЛ – конвеєрна лінія;
ОТ – операційні технології;
ПЗ – програмне забезпечення;
ПЛК – програмований логічний контролер;
СКБД – система керування базами даних;
ТЗ – технічне завдання;
ТО – технічне обслуговування;
API – Application Programming Interface;
CPU – Central Processing Unit;
DOM – Document Object Model;
ERP – Enterprise Resource Planning;
GUI – Graphical User Interface;
HMI – Human-Machine Interface;
HTTP – HyperText Transfer Protocol;
JS – JavaScript;
JSON – JavaScript Object Notation;
KPI – Key Performance Indicators;
MES – Manufacturing Execution System;
MTBF – Mean Time Between Failures;
MTTR – Mean Time To Repair;
MUI – Material User Interface;
OEE – Overall Equipment Effectiveness;
ORM – Object-Relational Mapping;

PDF – Portable Document Format;
PKCS – Public-Key Cryptography Standards;
PLC – Programmable Logic Controller;
RAM – Random Access Memory;
REST – Representational State Transfer;
SCADA – Supervisory Control And Data Acquisition;
SPA – Single Page Application;
SQL – Structured Query Language;
SSD – Solid-State Drive;
SSE – Server-Sent Events;
TCP/IP – Transmission Control Protocol / Internet Protocol;
UDP – User Datagram Protocol;
UI – User Interface;
UPS – Uninterruptible Power Supply;
VLAN – Virtual Local Area Network;
WAL – Write-Ahead Logging.

ВСТУП

Сучасний світ промислового виробництва постійно вимагає підвищення ефективності, надійності та безпеки технологічних процесів. Однією з ключових вимог є автоматизація управління виробничими одиницями, зокрема конвеєрними лініями (КЛ), що потребує розроблення сучасних та надійних систем контролю. Особливо актуальним це стає в умовах, де необхідний оперативний моніторинг численних параметрів та швидке реагування на відхилення від технологічних норм.

У зв'язку з цим виникає потреба у розробленні спеціалізованих інструментів для управління виробничими процесами. Особливою увагою користуються програмні комплекси, які дозволяють реалізувати Людино-машинний Інтерфейс та забезпечити достовірний контроль і документування стану обладнання.

Метою роботи є підвищення надійності та оперативності керування конвеєрною лінією шляхом розроблення спеціалізованого програмного комплексу моніторингу, що забезпечує аналіз дрейфу технологічних параметрів та діагностику стану обладнання.

У роботі буде здійснено аналіз існуючих підходів до автоматизації КЛ, проведено обґрунтування вибору програмно-технологічного стеку (JavaScript, React, Node.js, Electron) та розроблено прототип системи, що демонструє її функціональність у різних технологічних умовах.

Для розроблення прототипу системи необхідно формалізувати вхідні дані (статичні та динамічні), розробити алгоритми обчислення керуючих параметрів та оцінити якість системи за такими критеріями, як швидкість обробки даних, надійність архівування та зручність інтерфейсу.

Внесок у Цілі сталого розвитку: Розроблення даної автоматизованої системи управління конвеєрною лінією безпосередньо сприяє досягненню кількох Цілей сталого розвитку. Зокрема, вона робить внесок у Ціль 9:

Індустріалізація, інновації та інфраструктура, шляхом створення стійкої інфраструктури, сприяння інклюзивній та стійкій індустріалізації та заохочення інновацій (через застосування сучасного стеку Full-Stack JS та алгоритмів предиктивної діагностики). Крім того, завдяки підвищенню надійності виробничих процесів, система підтримує Ціль 12: Відповідальне споживання та виробництво, мінімізуючи прості обладнання та оптимізуючи використання ресурсів. Аналітичні модулі роботи, що прогнозують несправності, також дозволяють перейти від реактивного до превентивного обслуговування, продовжуючи термін служби обладнання.

До основних задач, які вирішує система, відносяться такі ключові аспекти:

- швидке та точне керування технологічними параметрами;
- автоматизація процесів моніторингу та контролю;
- адаптивність інтерфейсу до різних умов експлуатації;
- інтеграція з локальними системами передачі даних;
- масштабованість та продуктивність програмного забезпечення.

Об'єктом дослідження є процес управління технологічними режимами та обладнанням конвеєрної лінії, що включає збір, аналіз даних та формування керуючих впливів.

Предметом дослідження є програмний комплекс АСУ конвеєрною лінією, методи структурування даних, алгоритми виявлення дрейфу параметрів, принципи побудови НМІ та підсистеми протоколювання.

В роботі запропоновано програмний комплекс поточного контролю конвеєрних ліній, включаючи аналіз швидкості обробки даних, обробка дрейфів та рівня відповідності вимогам. ПЗ дозволяє інтегрувати модулі у гібридний застосунок із застосуванням НМІ, який може протестувати отриманий застосунок та розробити рекомендації щодо подальшого вдосконалення, що можна віднести до отриманих наукових результатів.

Для досягнення поставленої мети необхідно виконати наступні завдання:

- провести аналіз сучасних методів та технологій для автоматизації конвеєрних ліній та обґрунтувати вибір програмного стеку;
- дослідити вимоги до структури ПЗ, архівації та документування;
- формалізувати та підготувати структуру вхідних (статичних/динамічних) даних;
- розробити та реалізувати модулі Обчислення, Керування та Архівації даних;
- провести тестування та оцінку надійності розробленого ПЗ, включаючи аналіз швидкості обробки даних та рівня відповідності вимогам;
- інтегрувати модулі у гібридний застосунок із застосуванням НМІ;
- протестувати отриманий застосунок та розробити рекомендації щодо подальшого вдосконалення;
- оформити кваліфікаційну роботу згідно [1-2].

Результати роботи опубліковані в [3].

1 АНАЛІЗ ІСНУЮЧИХ ПІДХОДІВ ДО АВТОМАТИЗАЦІЇ КОНВЕЄРНИХ ЛІНІЙ

Сучасне виробництво характеризується посиленням вимог до гнучкості, ефективності та прозорості технологічних процесів. У цьому контексті автоматизація конвеєрних ліній перестала бути завданням виключно механічного впровадження, перетворившись на складну проблему інтеграції апаратних засобів, програмного забезпечення та комунікаційних протоколів. Даний розділ присвячено комплексному аналізу існуючих архітектурних підходів, комунікаційних платформ та програмних рішень для побудови АСУ конвеєрними лініями, з метою обґрунтування вибору оптимального технологічного стеку для розробки високоефективної та економічної системи.

1.1 Використання промислових контролерів та SCADA-систем

Автоматизована система управління конвеєрною лінією є складною ієрархічною розподіленою системою в архітектурі виробництва, що поєднує в собі засоби нижчого (польового) та верхнього (диспетчерського) рівнів. Фундаментом та "виконавчим мозком" такої системи виступають програмовані логічні контролери (ПЛК), які забезпечують безпосереднє виконання програм керування виконавчими механізмами (частотними перетворювачами, електромагнітними клапанами, сервоприводами) та збір первинних даних з датчиків і сенсорів. На цьому рівні реалізується детермінована логіка реального часу, що гарантує безпеку та стабільність технологічного процесу.

Інформаційно-керуючим інтерфейсом між технологічним процесом і людиною-оператором слугують SCADA-системи (Supervisory Control And Data Acquisition) або спеціалізовані засоби людино-машинного інтерфейсу. НМІ виступає як критично важлива ланка, що забезпечує візуалізацію стану

технологічного процесу в реальному часі, реєстрацію аварійних та технологічних тривог, формування звітів і надання можливості оперативного втручання персоналу. Ефективність роботи оператора повністю залежить від швидкодії, інтуїтивності та інформаційної насиченості НМІ.

Для ефективного функціонування такої розподіленої АСУ КЛ, комунікаційна підсистема має надійно обслуговувати чотирьома основними, принципово різними за вимогами, потоки трафіку:

Дані керування (Control Data): високопріоритетний, строго циклічний трафік реального часу, що вимагає детермінованої (передбачуваної) та наднизької затримки. Наприклад, сигнали аварійного зупину (E-Stop) або команди синхронізації приводів.

Дані моніторингу та візуалізації (HMI/SCADA Data): критичний трафік для відображення графіків, трендів реального часу та історичних даних, який характеризується передачею великих обсягів даних (стани сотень тегів, графічні об'єкти). Затримки в цьому потоці безпосередньо впливають на якість взаємодії оператора з системою.

Дані конфігурації/діагностики: ациклічний трафік для обслуговування системи, завантаження програм у ПЛК, зчитування логів та діагностики обладнання. Не вимагає мілісекундних затримок, але повинен бути стійким до перешкод.

Дані вертикальної інтеграції (MES/ERP Data): передача узагальнених виробничих даних (виробіток за зміну, загальний коефіцієнт корисної дії обладнання – OEE) у корпоративні системи верхнього рівня (MES, ERP) для планування та аналізу.

Традиційні промислові мережі (Fieldbus), такі як Modbus RTU (послідовний інтерфейс RS-485) та Profibus-DP, що домінували в минулому, сьогодні виявляють фундаментальні недоліки, які унеможливають їхнє використання у сучасних високовимогливих проектах. Їхніми ключовими обмеженнями є критично низька швидкість (до 115,2 кбіт/с для Modbus RTU) та обмежений розмір кадру (зазвичай ≈ 256 байт). Це змушує виконувати

множинні транзакції "запит–відповідь" для передачі об'ємних НМІ–даних, що призводить до значного зниження продуктивності, збільшення завантаження центрального процесора ПЛК та, як наслідок, підвищення ризику втрати критично важливих даних керування.

1.2 Обґрунтування вибору комунікаційної платформи TCP/IP

Для забезпечення високої швидкості, детермінованості та функціональності сучасного НМІ необхідний перехід на універсальний стек протоколів TCP/IP, реалізований на фізичному рівні через промисловий Ethernet [5]. Метою даного дослідження є обґрунтування переваг TCP/IP як транспортного рівня для високоінтегрованої АСУ конвеєра з НМІ, зокрема, через кількісне порівняння продуктивності з успадкованими рішеннями.

1.2.1 Кількісне обґрунтування часу затримки (Latency) НМІ

Критичним параметром для якості НМІ є час затримки (Latency) між зміною величини в ПЛК та її відображенням на екрані оператора. Кількісний розрахунок при типовому завантаженні НМІ–екрану, що вимагає передачі 1250 байт корисних даних (стани дискретних сигналів, значення аналогових змінних, строкові мітки), демонструє наступне порівняння:

– сценарій 1: Industrial Ethernet (TCP/IP): висока пропускна здатність 100 Мбіт/с дозволяє передати весь необхідний обсяг даних в одному пакеті Modbus TCP розміром 1315 байт (з урахуванням службових заголовків). Час передачі такого пакету становить $T \approx 0,105$ мс. Ця затримка є непомітною для людського сприйняття;

– сценарій 2: Fieldbus (Modbus RTU): низька швидкість (115,2 кбіт/с) та обмеження корисного навантаження кадру (≈ 256 байт, з яких лише ~ 250 байт доступні для даних Modbus) вимагають виконання 5 окремих транзакцій "запит–відповідь". Загальний час затримки, враховуючи накладні витрати на обробку кожного запиту, становить $T \approx 155,5$ мс.

Висновок: використання TCP/IP забезпечує оновлення НМІ–даних приблизно у 1500 разів швидше порівняно з класичним Modbus RTU. Затримка у 155,5 мс є візуально помітною для оператора і сприймається як "лаг", "підвисання" інтерфейсу, що значно знижує ефективність контролю за швидкоплинними процесами.

1.2.2 Декомпозиція стеку та гібридний транспортний підхід

Основним архітектурним перевагою стека TCP/IP є його модульність. Він забезпечує глобальну IP–адресацію (мережевий рівень), що дозволяє легко інтегрувати пристрої в єдину мережу, та гнучкість на транспортному рівні за рахунок вибору між протоколами TCP та UDP. Незважаючи на більші накладні витрати заголовка Modbus TCP/IP (65 байт), ніж у Modbus RTU (4 байти) (рис. 1.1), це повністю компенсується на декілька порядків вищою швидкістю передачі, що робить питому вагу службових даних у загальному трафіку незначною.

На транспортному рівні доцільним є використання гібридного підходу, що дозволяє оптимально розподілити трафік за пріоритетами:

- протокол TCP (Transmission Control Protocol): протокол із встановленням з'єднання та механізмом підтвердження доставки. Забезпечує гарантовану доставку та цілісність даних. Використовується для передачі критичних команд (наприклад, аварійна зупинка, запуск/зупинка лінії, зміна уставок), де пропуск команди неприпустимий;

- протокол UDP (User Datagram Protocol): протокол без встановлення з'єднання і без підтвердження доставки. Використовується для поточкових даних моніторингу (тренди НМІ, масиви миттєвих значень), де важлива актуальність останніх даних, а не абсолютна цілісність усіх пакетів. Втрата окремого пакету в потоці не є критичною, а використання UDP мінімізує джиттер (варіації затримки) та накладні витрати.

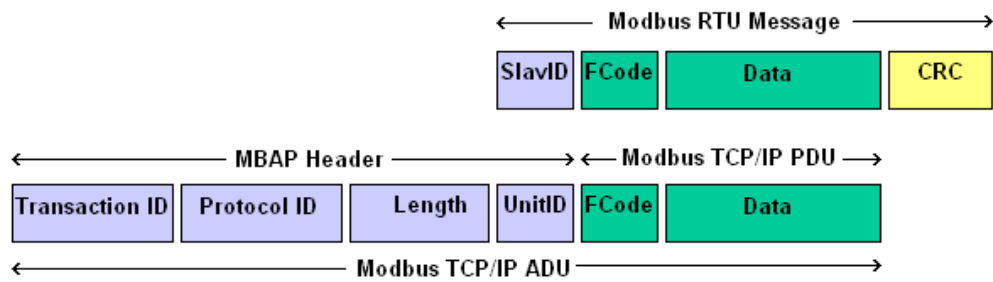


Рисунок 1.1 – Структура пакету Modbus TCP/IP та Modbus RTU

1.3 Аналіз комерційних рішень та обґрунтування вибору технологічного стеку (Full-Stack JS)

Комерційний ринок АСУ КЛ представлений зрілими, високонадійними, але переважно закритими (пропрієтарними) рішеннями, які часто характеризуються високою вартістю володіння та обмеженою гнучкістю. Це створює потужний аргумент на користь розробки власного програмного забезпечення на базі сучасних відкритих веб-технологій, що дозволяє отримати конкурентні переваги [4].

1.3.1 Детальний аналіз існуючих комерційних систем

Найбільш поширеними комерційними рішеннями для HMI/SCADA рівня є системи, що базуються на пропрієтарних стандартах та жорстко інтегровані з апаратним забезпеченням конкретного виробника [18, 20]:

- siemens TIA Portal (SIMATIC HMI та WinCC): ця платформа є промисловим стандартом для автоматизації. Вона використовує власний протокол Profinet (реалізований на базі TCP/IP), що забезпечує високу інтеграцію апаратного та програмного забезпечення Siemens. Проте, системи мають високу вартість ліцензування (особливо для розподілених систем WinCC), створюють повну залежність від екосистеми Siemens та мають суттєві обмеження у кастомізації інтерфейсу під специфічні вимоги замовника;

- rockwell Automation (FactoryTalk View SE): аналогічна за потужністю та закритістю платформа, що використовує протокол EtherNet/IP. Має жорстку

прив'язку до апаратного забезпечення Allen–Bradley, вимагає придбання окремих ліцензій на сервери, клієнтські місця та продукти розробки, що значно збільшує капітальні витрати;

– *ignition by Inductive Automation*: відносно сучасне рішення, що використовує веб–технології та модель ліцензування на основі підписки. Його перевагою є крос–платформність, проте вартість ліцензії залишається значною для малих та середніх проектів, а глибока кастомізація часто вимагає знань мови програмування Java.

Недоліки цих систем, пов'язані з високою вартістю, "вендор–локом" (залежністю від постачальника) та обмеженою гнучкістю, можуть бути усунені завдяки розробці власного рішення на базі відкритих технологій.

1.3.2 Переваги розроблюваної системи на базі Full–Stack JavaScript

Розробка власного НМІ на базі стеку JavaScript (Full–Stack JS) [14], зокрема React, Node.js та Electron, пропонує комплекс економічних, технічних та ергономічних переваг, усуваючи ключові недоліки пропрієтарних систем:

– бібліотека React.js (Клієнтський інтерфейс): Ця бібліотека ідеально підходить для створення складних, компонентно–орієнтованих НМІ [4]. Вона дозволяє розробляти перевикористовувані графічні компоненти (кнопки, індикатори, графіки трендів), що суттєво прискорює розробку та полегшує підтримку. Віртуальний DOM (Document Object Model) та ефективні алгоритми реактивного оновлення інтерфейсу забезпечують швидке та плавне відображення даних в реальному часі, що критично важливо для якісного НМІ;

– середовище Node.js (серверна логіка / шлюз даних): середовище виконання JavaScript на стороні сервера [5]. Воно високоефективне для I/O операцій та мережевих з'єднань завдяки неблокуючій, асинхронній моделі. Це робить Node.js ідеальним вибором для реалізації шлюзу, який одночасно обслуговує численні підключення до ПЛК (за протоколами Modbus TCP, OPC UA), збирає дані, агрегує їх і надає через API (REST, WebSocket) клієнтському додатку на React;

– фреймворк Electron (Платформа для розгортання): фреймворк для створення крос-платформних настільних додатків з використанням веб-технологій [6]. Він дозволяє "упакувати" додаток React разом із сервером Node.js в єдиний самостійний виконуваний файл (рис. 1.2). Це забезпечує зручне локальне розгортання на робочому місці оператора без необхідності встановлення додаткового ПЗ (браузера, сервера) і забезпечує повний контроль над середовищем виконання.

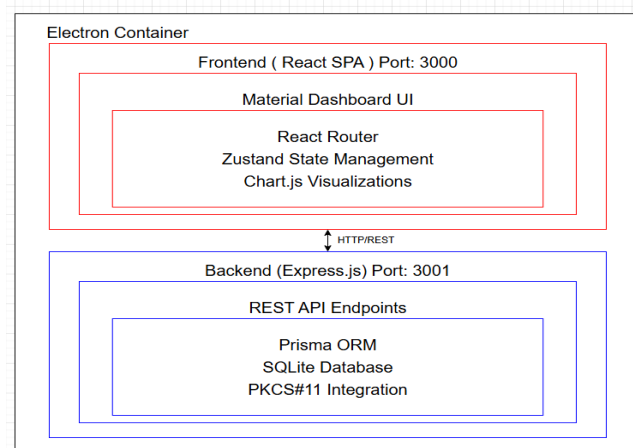


Рисунок 1.2 – Архітектура програмного забезпечення на базі Full-Stack JS (React + Node.js)

1.4 Аналіз методів архівації та документування даних в АСУ

Сучасна АСУ КЛ не обмежується функціями керування в реальному часі. Ефективне архівування історичних даних, документування подій та формування звітів є невід'ємною частиною виробничого процесу, що забезпечується завдяки вертикальній інтеграції, яку надає TCP/IP. Ця інтеграція є технологічною основою для конвергенції операційних (OT) та інформаційних (IT) технологій.

1.4.1 Вертикальна інтеграція та роль НМІ в архітектурі

Вертикальна інтеграція – це пряма, безшовна взаємодія між цеховим (OT, рівень ПЛК та НМІ) та корпоративним (IT, рівень MES/ERP) рівнями

управління. TCP/IP як універсальна транспортна основа дозволяє НМІ–серверу або шлюзу даних безпосередньо обмінюватися структурованими даними з системами MES (Manufacturing Execution System) та ERP (Enterprise Resource Planning). Це усуває потребу в дорогих та складних проміжних шлюзах–конверторах протоколів. НМІ в такій архітектурі виступає не лише як інтерфейс оператора, але і як агрегатор даних: він збирає сирі дані з ПЛК, проводить їх первинну обробку (наприклад, розрахунок ОЕЕ) і передає в системи верхнього рівня узагальнені виробничі показники (виробіток за зміну, час простоїв, причини відхилень, коди помилок обладнання) для довгострокового архівування та глибшого аналізу.

1.4.2 Системи документування, протоколювання та вимоги до кібербезпеки

Для забезпечення відстеженості виробничого процесу АСУ КЛ повинна підтримувати два ключових види документування:

– протоколювання (Logging): Безперервний запис усіх значущих подій системи в хронологічному порядку. Це включає аварійні та попереджувальні сигнали, дії оператора (включення, зупинка, зміна уставок), зміни режимів роботи, а також помилки в комунікації. Ці логи необхідні для проведення розслідування інцидентів та аналізу причин простоїв;

– звітність (Reporting): Автоматичне формування регламентованих звітів (змінних, добових, місячних) на основі архівованих даних. Звіти можуть включати дані про виконані завдання, витрату матеріалів, продуктивність обладнання тощо.

Однак, перехід на відкритий стандарт TCP/IP та інтеграцію з корпоративними мережами різко підвищує ризики кібербезпеки [15, 19]. Для забезпечення надійності та безпеки функціонування АСУ КЛ необхідний багаторівневий захист ("Defense in Depth"): обов'язкове сегментування мережі за допомогою VLAN, використання демілітаризованої зони для контрольованого та безпечного обміну даними між ОТ та ІТ сегментами, а

також налаштування міжмережевих екранів (Firewall) з жорсткими правилами фільтрації трафіку, що блокує несанкціонований доступ до критичної інфраструктури [17].

1.5 Висновки до розділу

Проведений аналіз підтверджує, що сучасні вимоги до продуктивності, функціональності та економічної ефективності АСУ КЛ можуть бути повною мірою задоволені лише через послідовне використання двох ключових технологій: Industrial Ethernet/TCP/IP на комунікаційному рівні та Full-Stack JavaScript на програмному.

Кількісне порівняння демонструє, що перехід на TCP/IP мінімізує затримку оновлення НМІ-інтерфейсу, роблячи його роботу практично миттєвою (швидкість передачі зростає приблизно у 1500 разів порівняно з застарілим Modbus RTU). Гнучкість стеку дозволяє реалізувати гібридний транспортний підхід (TCP для команд, UDP для потокових даних), оптимізуючи навантаження на мережу.

Вибір технологічного стеку на базі React, Node.js та Electron для розробки власного ПЗ дозволяє усунути фундаментальні недоліки пропрієтарних SCADA/НМІ-систем: високу вартість ліцензування, залежність від вендора та обмежену кастомізацію. Цей підхід забезпечує створення високоефективного, масштабованого та економічно вигідного програмного забезпечення для АСУ КЛ, що повною мірою відповідає концепціям Індустрії 4.0 та цифрової трансформації виробництва.

2 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ ТА ПОСТАНОВКА ЗАДАЧ ДОСЛІДЖЕННЯ

Перехід від теоретичного аналізу сучасних тенденцій автоматизації до практичної реалізації потребує чіткого визначення цілей та рамок проекту. Даний розділ виконує роль зв'язуючої ланки між аналітичним оглядом і безпосередньою розробкою, ґрунтуючись на вимогах Технічного завдання (ТЗ). Основним завданням розділу є систематизація, структуризація та наукова формалізація вимог ТЗ, їх аналіз на предмет взаємозв'язків та потенційних протиріччій, а також трансформація цих вимог у чіткий переклад науково – технічних завдань. Результатом цієї роботи стає концептуальна основа для проектування архітектури та вибору оптимальних методів реалізації програмного комплексу АСУ КЛ.

2.1 Аналіз вихідних умов та призначення системи

Відповідно до ТЗ та стандарту вищої освіти, розроблювана система є спеціалізованим програмним рішенням, інтегрованим у фізичне робоче місце оператора конвеєрної лінії. Ключовою характеристикою системи є її локалізованість та автономність. Вона не є розподіленою SCADA–системою цехового рівня, а представляє собою локальну однокористувацьку АСУ, що функціонує в рамках єдиного персонального комп'ютера. Це обмеження формує специфічні вимоги до архітектури, зокрема, відмову від клієнт–серверної моделі з мережевими взаємодіями на користь гібридного додатка, що інкапсулює всі компоненти.

Фундаментальним викликом, який вирішує система, є реалізація процесу безперервного технологічного моніторингу та активного контролю. На відміну від пасивних систем спостереження, розроблювана АСУ КЛ має не лише фіксувати дані, але й проводити їх аналіз за складними алгоритмами для прогностичної оцінки стану обладнання. Це реалізується через концепцію

контролю груп однотипних датчиків, що вимірюють один технологічний параметр. Такий підхід, заснований на принципі порівняння, дозволяє виявляти аномалії та деградацію характеристик окремих датчиків без необхідності їх демонтажу для перевірки, що є істотною перевагою для безперервних виробничих процесів.

Система повинна адекватно реагувати на полірежимність технологічного процесу, що вимагає реалізації різних моделей аналізу:

- аналіз у стаціонарному режимі: сприяє виявленню низькочастотних трендів, таких як повільний "дрейф" метрологічних характеристик датчиків, що може бути ознакою їх старіння або забруднення [19];

- аналіз у динамічному режимі: дозволяє оцінити лінійність та точність датчиків в усьому робочому діапазоні, виявляючи нелінійні спотворення та зони нечутливості, які не проявляються при сталому значенні параметра.

2.2 Формалізація функціональних вимог до ПЗ АСУ КЛ

Системний аналіз ТЗ дозволяє класифікувати функціональні вимоги на кілька взаємопов'язаних груп, що формують ярусну модель функціональності системи.

2.2.1 Вимоги до модуля збору даних та управління:

- забезпечення підключення до джерел даних (ПЛК, файлові архіви) через стандартні промислові протоколи (Modbus TCP) або шляхом імпорту структурованих файлів (наприклад, .xlsx);

- реалізація механізму реального часу для читання та первинної обробки вхідних сигналів з мінімальною затримкою;

- надання інструментів для оперативного керування виконавчими механізмами КЛ (запуск, зупинка, аварійне відключення, зміна заданих уставок) з обов'язковим підтвердженням виконання команд.

2.2.2 Вимоги до модуля аналітичної обробки даних:

Реалізація алгоритмів статистичного аналізу для груп датчиків, зокрема:

- розрахунок середньогруппового значення з адаптивною фільтрацією "викидів";
- кваліметричний аналіз – обчислення миттєвих та середніх відхилень кожного датчика від еталонного (середньогруппового) значення;
- обчислення дрейфу характеристик датчиків як різниці середніх значень між різними часовими періодами стаціонарного режиму;
- реалізація логіки прийняття рішень на основі аналізу – автоматичне порівняння отриманих відхилень та дрейфу з технологічно допустимими межами, що задаються в конфігурації.

2.2.3 Вимоги до системи візуалізації та НМІ:

- створення інтуїтивного графічного інтерфейсу, що надає оператору єдину точку контролю за станом КЛ;
- візуалізація у формі мнемосхем, графіків реального часу, трендів, таблиць стану та кольорового кодування аварійних станів;
- забезпечення можливості гнучкого конфігурування відображуваних даних і груп датчиків без зміни програмного коду.

2.2.4 Вимоги до підсистеми архівування та звітності:

- організація надійного та ефективного зберігання великих масивів історичних даних (сирі покази, результати аналізу, системні події);
- реалізація механізму формування звітів різного рівня деталізації – від оперативних звітів за зміну до комплексних аналітичних звітів за цикл роботи;
- забезпечення експорту звітів у промислово–стандартні формати (.xlsx, .pdf) для подальшого використання в системах верхнього рівня (MES, ERP).

2.2.5 Вимоги до адміністративних та сервісних функцій:

- реалізація системи обліку користувачів з розмежуванням прав доступу

(оператор, інженер, адміністратор);

- ведення детального журналу подій (лог-файлу) з фіксацією всіх значущих дій оператора, змін конфігурації та системних помилок;
- надання інструментів для резервного копіювання та відновлення конфігураційних і архівних даних.

2.3 Аналіз нефункціональних вимог та обмежень

Нефункціональні вимоги визначають якісні характеристики системи та накладають суттєві обмеження на архітектурні рішення.

Вимоги до архітектури та середовища виконання: жорстка прив'язка до ОС Windows та модель однокористувацького локального застосунку вимагають вибору технологій, що дозволяють створити самостійний гібридний додаток. Це робить технологічний стек на базі Electron, Node.js та React оптимальним вибором, оскільки він дозволяє інкапсулювати back-end, front-end та базу даних в єдиний виконуваний файл, усуваючи залежність від зовнішніх веб-серверів та браузерів.

Вимоги до продуктивності: інтерфейс користувача (НМІ) повинен забезпечувати оновлення даних з частотою, достатньою для комфортного сприйняття оператором (мінімум 1–2 рази на секунду для графіків трендів). Це вимагає ефективної реалізації механізмів обміну даними між модулями та оптимізації роботи з графікою.

Вимоги до надійності: система повинна бути стійкою до помилок вводу-виводу, таких як обрив зв'язку з ПЛК або пошкодження файлів для імпорту. Критичні функції керування повинні мати захист від несанкціонованого доступу та помилок оператора.

Вимоги до безпеки: необхідність протоколювання та автентифікації вимагає реалізації механізмів кібербезпеки на рівні застосунку, включаючи шифрування паролів та контроль цілісності лог-файлів.

2.4 Постановка науково–дослідних та прикладних задач

На основі глибокого аналізу функціональних та нефункціональних вимог формулюються наступні задачі дослідження та розробки, що становлять програмний каркас проекту:

- розробити оптимізовану архітектуру програмного комплексу АСУ КЛ для гібридного десктопного застосування, що забезпечить ефективну взаємодію між модулем збору даних, аналітичним ядром, системою керування, НМІ та базою даних в умовах обмежених апаратних ресурсів;

- спроектувати та реалізувати реляційну схему бази даних, що забезпечить цілісне зберігання гетерогенних даних: статичних (конфігурація датчиків, груп, користувачів) та динамічних (архів показів, результати розрахунків, журнал подій), із забезпеченням швидкості вибірки для побудови трендів та звітів;

- розробити та верифікувати алгоритми аналітичної обробки даних, що включають методи статистичного аналізу груп датчиків, алгоритми виявлення та компенсації "викидів", математичні моделі для розрахунку дрейфу та відхилень з урахуванням різних технологічних режимів (стаціонарний, динамічний);

- реалізувати модуль НМІ з використанням сучасного GUI–фреймворку (React), що надасть зручні інструменти для візуалізації стану КЛ в реальному часі, гнучкого конфігурування системи, формування запитів до архіву та генерації звітів різного ступеня складності;

- інтегрувати підсистему безпеки та аудиту, що включає механізми автентифікації користувачів, розмежування прав доступу на рівні функцій та даних, а також централізоване протоколювання всіх значущих подій для забезпечення відповідальності та аналізу інцидентів;

- провести комплексну верифікацію та валідацію розробленого ПЗ шляхом створення тестового середовища, що імітує роботу КЛ, з подальшим тестуванням на відповідність функціональним вимогам ТЗ, перевіркою

продуктивності та стрес–тестуванням для оцінки стабільності роботи в умовах високого навантаження.

2.5 Висновки до розділу

Аналіз ТЗ дозволив не лише перелічити вимоги, але й виявити їхню внутрішню логіку та взаємозалежність. Встановлено, що ключовою відмінністю розроблюваної системи є її орієнтація на глибокий аналіз даних за спеціальними алгоритмами в рамках автономного робочого місця, що відрізняє її від типових SCADA–систем.

Сформульовані шість основних задач утворюють послідовний і взаємопов'язаний ланцюг робіт, що охоплює всі етапи життєвого циклу розробки ПЗ – від системного проектування до експериментальних досліджень. Вирішення цих задач дозволить створити не просто програмний продукт, що відповідає ТЗ, а конкурентоспроможне, науково обґрунтоване рішення, здатне підвищити ефективність, надійність та безпеку управління конвеєрною лінією в умовах сучасного виробництва.

3 АРХІТЕКТУРА ТА ПРОЕКТУВАННЯ СИСТЕМИ КЕРУВАННЯ КОНВЕЄРНОЮ ЛІНІЄЮ

3.1 Функціональні вимоги до ПЗ АСУ КЛ

Проектування програмного забезпечення (ПЗ) для автоматизованої системи управління конвеєрною лінією базується на детальному аналізі технологічних процесів та потреб оперативного персоналу. Ключовою метою є створення надійного інструменту, який поєднує функції моніторингу, керування та аналізу даних у єдиному інтерфейсі.

3.1.1 Вимоги до керування та моніторингу в реальному часі

Система повинна забезпечувати безперервний збір та візуалізацію даних з контрольних точок конвеєрної лінії. Це передбачає відображення поточних значень технологічних параметрів, стану датчиків та виконавчих механізмів з мінімальною затримкою. Інтерфейс оператора має надавати можливість не лише спостерігати за процесом, але й здійснювати керуючий вплив, наприклад, змінювати режими роботи обладнання або виконувати аварійну зупинку. Важливим аспектом є система сповіщень, яка повинна миттєво реагувати на вихід параметрів за допустимі межі, фіксуючи порушення технологічних умов.

3.1.2 Вимоги до архівації даних та звітності

Критичною вимогою є збереження повної історії функціонування лінії. Програмний комплекс має забезпечувати транзакційний запис вимірювань у локальну базу даних, що гарантує цілісність інформації навіть у випадку збоїв живлення. Окрім автоматичного збору даних, система повинна підтримувати імпорт історичних даних із зовнішніх файлів (формати Excel/ТХТ), що дозволяє інтегрувати звіти з інших систем обліку. Підсистема звітності повинна генерувати аналітичні документи, включаючи графіки дрейфу

параметрів та статистику відхилень, з можливістю їх експорту у загальноприйнятні формати (PDF).

3.1.3 Вимоги до безпеки та надійності

Забезпечення цілісності та конфіденційності даних є пріоритетним завданням. Система повинна реалізовувати розмежування прав доступу на основі ролей, запобігаючи несанкціонованим змінам налаштувань або втраті критичних даних. Архітектура повинна передбачати механізми валідації вхідних даних на етапі їх отримання, щоб унеможливити потрапляння помилкових значень до бази даних.

3.2 Архітектура програмного комплексу

Для реалізації поставлених вимог було обрано архітектуру гібридного десктопного застосунку, побудованого на базі платформи Electron. Такий підхід дозволяє поєднати гнучкість веб-технологій з потужністю нативних додатків, забезпечуючи кросплатформеність та автономність роботи, на рисунку 3.1 можна побачити загальну архітектуру застосунку.

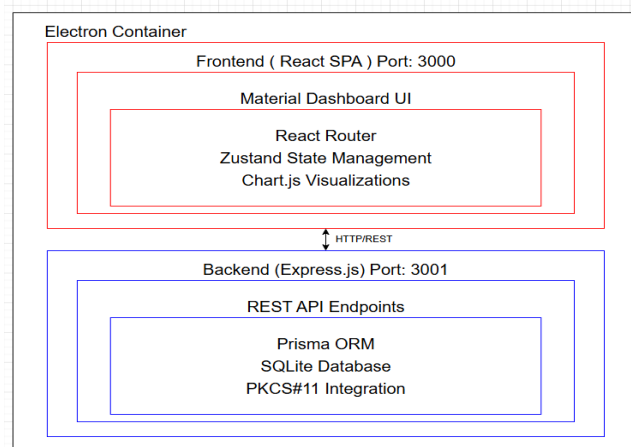


Рисунок 3.1 – Загальна архітектура

3.2.1 Фізична архітектура системи

Фізична реалізація системи передбачає розгортання програмного

комплексу на автоматизованому робочому місці оператора (АРМ). Система функціонує як автономний вузол, де всі компоненти – клієнтський інтерфейс, сервер обробки даних та система керування базами даних – інкапсульовані в межах одного фізичного пристрою. Зв'язок із зовнішнім світом (ПЛК або корпоративна мережа) здійснюється через мережеві інтерфейси ПК, а взаємодія з користувачем – через стандартні засоби вводу–виводу.

3.2.2 Логічна архітектура застосунку

Логічно застосунок розділений на три рівні, взаємодія між якими відбувається через чітко визначені інтерфейси:

- рівень представлення (Frontend): відповідає за взаємодію з користувачем, візуалізацію даних та формування запитів [13]. Реалізований як односторінковий додаток (SPA), що працює у середовищі Chromium всередині Electron;

- рівень бізнес–логіки (Backend): виконується у середовищі Node.js як окремий процес. Він обробляє API–запити, виконує математичні розрахунки, керує імпортом даних та забезпечує валідацію;

- рівень даних (Database): забезпечує персистентне зберігання інформації. Використовується вбудована реляційна база даних, що дозволяє зберігати конфігурацію та великі масиви часових рядів без необхідності встановлення окремого сервера БД.

3.2.3 Опис компонентів технологічного стеку

Вибір технологічного стеку обумовлений необхідністю використання єдиної мови програмування (JavaScript) для всіх рівнів системи, що спрощує розробку та підтримку.

- Frontend: побудований на бібліотеці React з використанням компонентної бази Material UI [10] для створення ергономічного інтерфейсу. Для керування станом застосунку використовується Zustand [12], а для візуалізації графіків – бібліотека Chart.js [11];

- Backend: реалізований на фреймворку Express.js [7], що працює у середовищі Node.js. Це забезпечує високу продуктивність при обробці I/O операцій та роботу з файловою системою;
- база даних: використовується SQLite [9] під управлінням ORM Prisma [8]. Prisma забезпечує типобезпечну взаємодію з БД, автоматичну генерацію міграцій та зручний API для виконання запитів;
- інтеграція: платформа Electron об'єднує всі компоненти в єдиний виконуваний файл, надаючи доступ до нативних API операційної системи.

3.3 Проектування структури даних

Ефективність системи керування залежить від правильно спроектованої моделі даних, яка повинна відображати фізичну структуру виробництва та забезпечувати швидкий доступ до результатів вимірювань.

3.3.1 Вибір та обґрунтування СКБД

В якості системи керування базами даних обрано SQLite. Це рішення є оптимальним для локальних систем керування, оскільки база даних зберігається у вигляді одного файлу на диску, не потребує адміністрування та налаштування мережевого доступу. Завдяки механізму WAL (Write-Ahead Logging), SQLite забезпечує високу швидкість запису даних, що є критичним для збереження потоків телеметрії. Використання Prisma ORM поверх SQLite дозволяє абстрагуватися від чистого SQL та працювати з даними як з об'єктами.

3.3.2 Опис основних сутностей бази даних

Структура бази даних розроблена для ієрархічного представлення конвеєрної лінії, їх можна побачити на рисунку 3.2.

підключаються Sensor (датчики) та Actuator (виконавчі механізми);

- дані вимірювань: для зберігання часових рядів використовуються таблиці Measurement (час вимірювання) та MeasurementData (конкретне значення параметра), що дозволяє ефективно зберігати великі обсяги історичних даних;

- користувачі: таблиця User містить облікові дані, а UserPreferences зберігає індивідуальні налаштування інтерфейсу.

3.4 Проектування програмних модулів

Програмна реалізація системи базується на модульному принципі, де кожен модуль відповідає за окремий аспект функціонування АСУ КЛ.

3.4.1 Модуль імпорту та валідації даних

Цей модуль забезпечує завантаження даних із зовнішніх джерел (Excel-файлів) для подальшого аналізу. Оскільки файли можуть бути великого розміру, реалізовано механізм потокової обробки та покрокової валідації. Основна логіка включає перевірку структури файлу, валідацію числових значень, дат та часу, а також перевірку на дублікати інтервалів у базі даних.

Для забезпечення інтерактивності прогрес імпорту передається клієнту через Server-Sent Events (SSE). Нижче наведено фрагмент коду, що відповідає за транзакційне збереження перевірених даних у базу даних за допомогою Prisma ORM:

```
// Фрагмент реалізації функції validateAndImportData
return await prisma.$transaction(async (tx) => {
  try {
    // Попередня валідація даних перед імпортом
    const validationErrors = validateDataBeforeImport();
    if (validationErrors.length > 0) {
```

```

// Логіка обробки та нормалізації помилок
throw { type: 'DATA_VALIDATION_FAILED', validationErrors, /.../
};

}

const measurementsMap = new Map();
// Створення вимірювань (Measurements)
for (let i = 0; i < validRowIndices.length; i++) {
  const row = validRowIndices[i];
  // Парсинг дати та часу
  const date = parseEuropeanDate(datesData[row][0]);
  const time = timesData[row][0].split(':').map(Number);
  date.setHours(...time)
  const measurement = await tx.measurement.create({
    data: {
      control_point_id: controlPoint.id,
      measurement_index: i + 1,
      measurement_date: date
    }
  });
  measurementsMap.set(row, measurement);
  // Відправка прогресу клієнту
  sendProgress(res, { stage: 'import', current: i, total:
validRowIndices.length * 2, /.../ });
}
// ... Збереження значень вимірювань (MeasurementData) для кожного
сенсора ...
return importResult;
} catch (error) {
  throw error;
}

```

});

Алгоритм роботи модуля передбачає декілька етапів перевірки: від перевірки наявності обов'язкових параметрів до валідації кожного окремого значення на відповідність допустимим діапазнам на рисунку 3.3 зображено блок–схему алгоритму роботи.

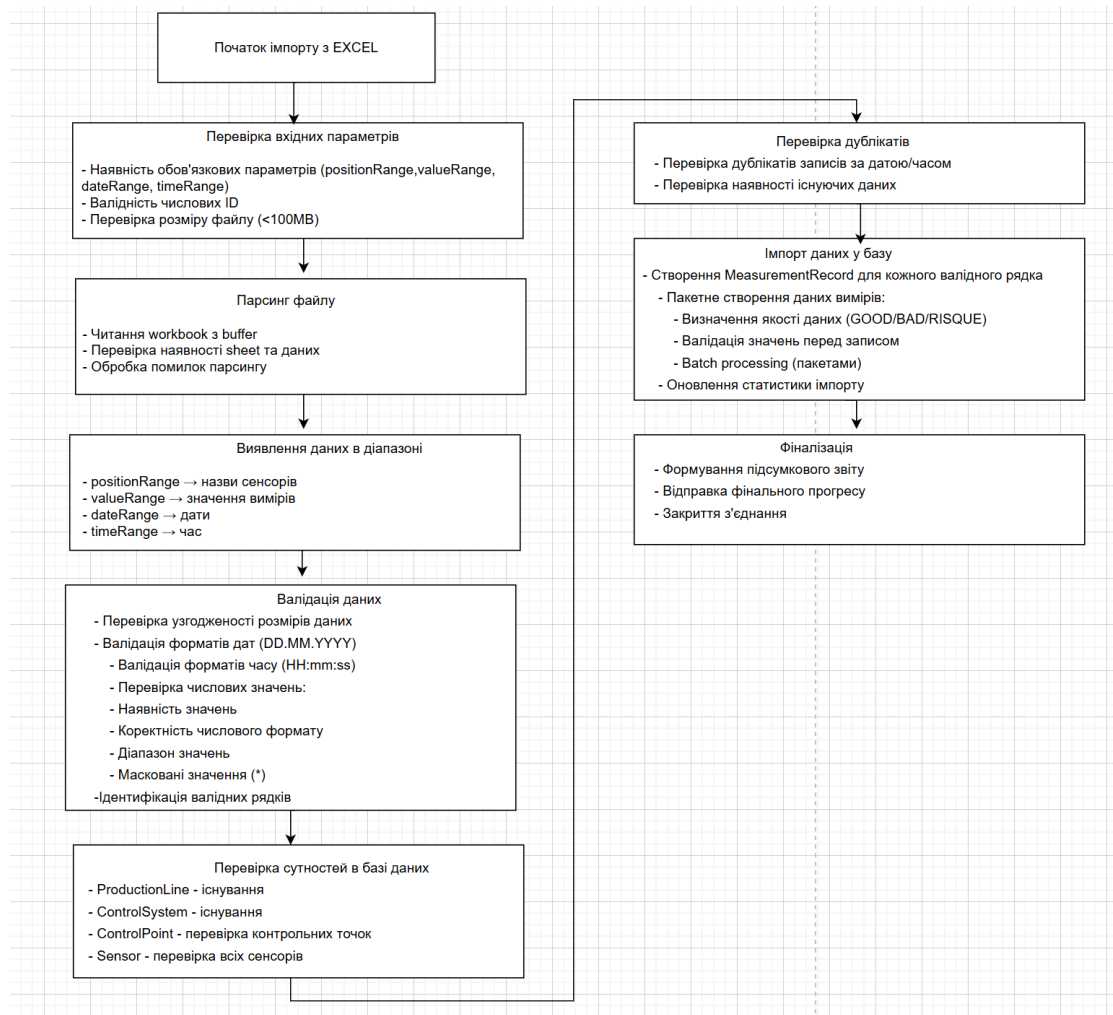


Рисунок 3.3 – Алгоритм імпорту та валідації даних з Excel

3.4.2 Модуль розрахунків та аналізу

Модуль розрахунків відповідає за бізнес–логіку обробки "сирих" даних. Він реалізований як окремий сервіс CalculatingService і надає API для підготовки даних, розрахунку відхилень та виявлення дрейфів технологічних параметрів. Архітектура модуля дозволяє виконувати як окремі етапи

розрахунку, так і комплексний аналіз за один запит.

Нижче на рисунку 3.4 наведено реалізацію маршрутизатора API, який обробляє запити на комплексний аналіз:

```
// Маршрут для комплексного аналізу даних
router.post("/analyze", async (req, res) => {
  try {
    const {
      selectedCycle, selectedInterval, selectedChannels,
      selectionCriteria = {}, preparationCriteria = {},
      driftDetectionParams = {}
    } = req.body;
    // Етап 1: Підготовка даних (нормалізація, фільтрація)
    const preparedData = await calculatingDataService.prepareData( {
      selectedCycle, selectedInterval, selectedChannels,
      selectionCriteria, preparationCriteria
    });
    // Етап 2: Обробка даних (розрахунок статистичних показників)
    const processedData = await
calculatingDataService.processData(preparedData);
    // Етап 3: Виявлення дрейфів (аналіз трендів)
    const driftAnalysis = await calculatingDataService.detectDrifts(
      preparedData, driftDetectionParams
    );
    // Етап 4: Формування звіту валідації
    const validationReport = await
calculatingDataService.getValidationReport(
      parseInt(selectedInterval),
      selectedChannels.map(id => parseInt(id))
    );
```

```

res.json({
  success: true,
  data: { preparedData, processedData, driftAnalysis, validationReport }
});
} catch (error) {
  console.error("Error in /api/calculating-data/analyze:", error);
  res.status(500).json({ success: false, error: error.message });
}
});

```

Така структура дозволяє гнучко налаштовувати параметри аналізу та отримувати деталізовані результати для візуалізації на графіках НМІ.

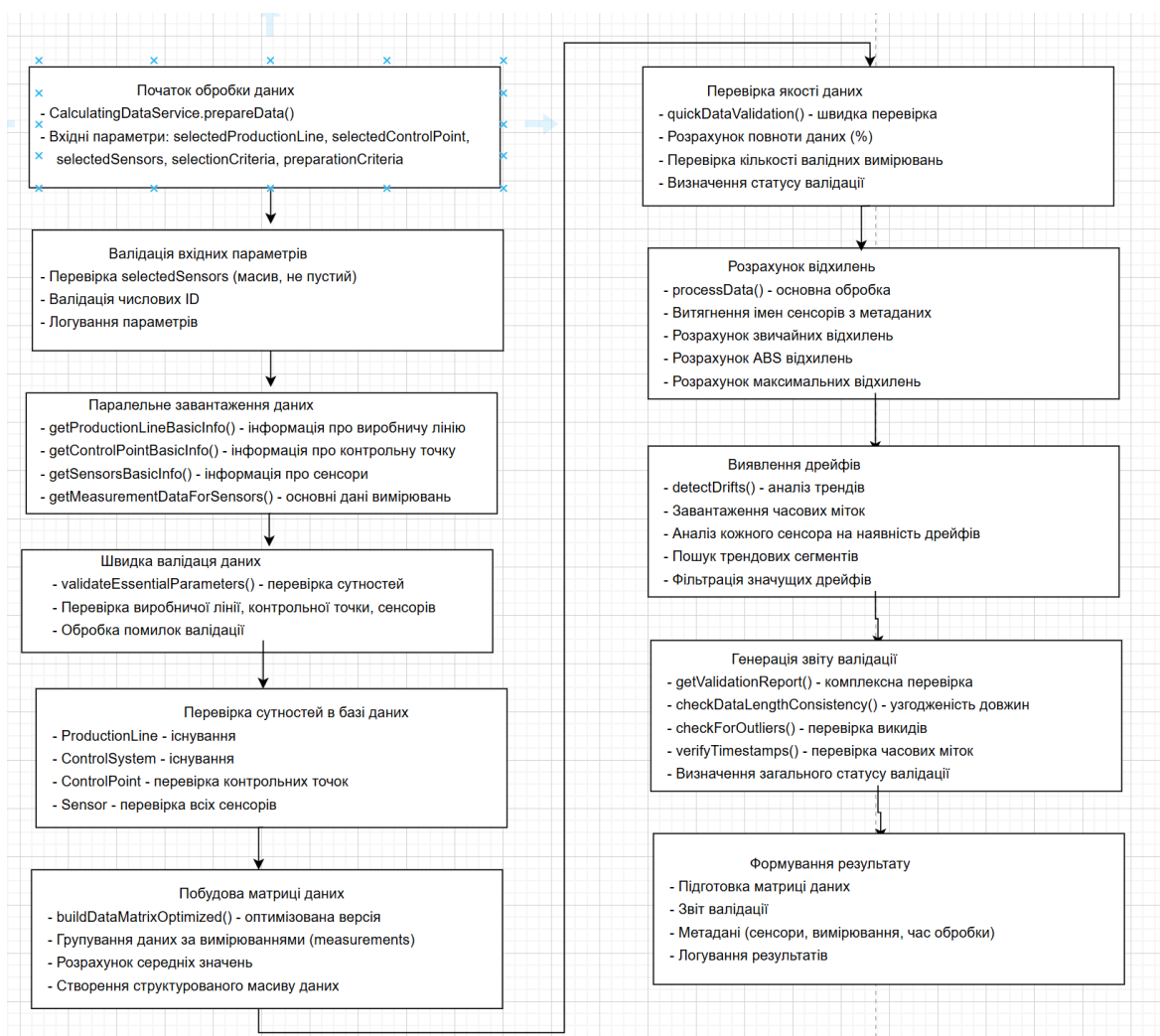


Рисунок 3.4 – Алгоритм розрахунку та виявлення дрейфів

3.4.3 Модуль НМІ та візуалізації

Модуль інтерфейсу відповідає за взаємодію з оператором. Він реалізує динамічне відображення таблиць та графіків. Використання бібліотеки React дозволяє створити реактивний інтерфейс, де зміни у стані системи (наприклад, завершення імпорту або надходження нових розрахунків) миттєво відображаються на екрані без перезавантаження сторінки. Компоненти візуалізації підтримують масштабування графіків та інтерактивний вибір часових інтервалів для детального аналізу

3.4.4 Модуль аутентифікації та безпеки

Модуль безпеки реалізує комплексний підхід до захисту інформаційної системи, забезпечуючи конфіденційність та цілісність даних. Процедура аутентифікації користувачів базується на використанні криптографічно стійкого алгоритму хешування bcrypt [15], що гарантує безпечне зберігання облікових даних і унеможливорює відновлення паролів навіть у випадку компрометації бази даних.

Управління сесіями реалізовано за допомогою бібліотеки express-session, яка забезпечує надійне зберігання стану авторизації між HTTP-запитами. Контроль доступу до функціональних можливостей системи здійснюється через ланцюжок middleware-компонентів (requireAuth, requireRole), які виконують перевірку наявності активної сесії та валідацію прав доступу користувача відповідно до його ролі перед виконанням будь-якої бізнес-логіки. Додатково, для підвищення рівня безпеки, система підтримує інтеграцію з апаратними носіями ключової інформації через інтерфейс PKCS#11, що дозволяє реалізувати сувору двофакторну аутентифікацію.

3.5 Висновки до розділу

У цьому розділі було розроблено архітектуру програмного забезпечення автоматизованої системи управління конвеєрною лінією. Обґрунтовано вибір трирівневої архітектури на базі стеку технологій JavaScript (React, Node.js, Electron) та бази даних SQLite, що забезпечує високу продуктивність, кросплатформеність та автономність рішення. Спроектвана структура бази даних дозволяє ефективно зберігати конфігураційні та технологічні дані. Розроблені алгоритми програмних модулів забезпечують надійний імпорт, валідацію та математичну обробку даних для моніторингу стану обладнання в реальному часі.

4 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ДОСЛІДЖЕННЯ ХАРАКТЕРИСТИК СИСТЕМИ

4.1 Реалізація інтерфейсу користувача та основних функцій

Розроблене програмне забезпечення являє собою кросплатформний десктопний додаток. Реалізація інтерфейсу виконана з дотриманням принципів ергономіки та Material Design, що забезпечує інтуїтивно зрозумілу взаємодію оператора з системою керування конвеєрною лінією.

4.1.1 Автентифікація та головна панель

Для захисту від несанкціонованого доступу вхід у систему здійснюється через форму автентифікації (рис. 4.1). Реалізовано механізм перевірки облікових даних з використанням хешування паролів.

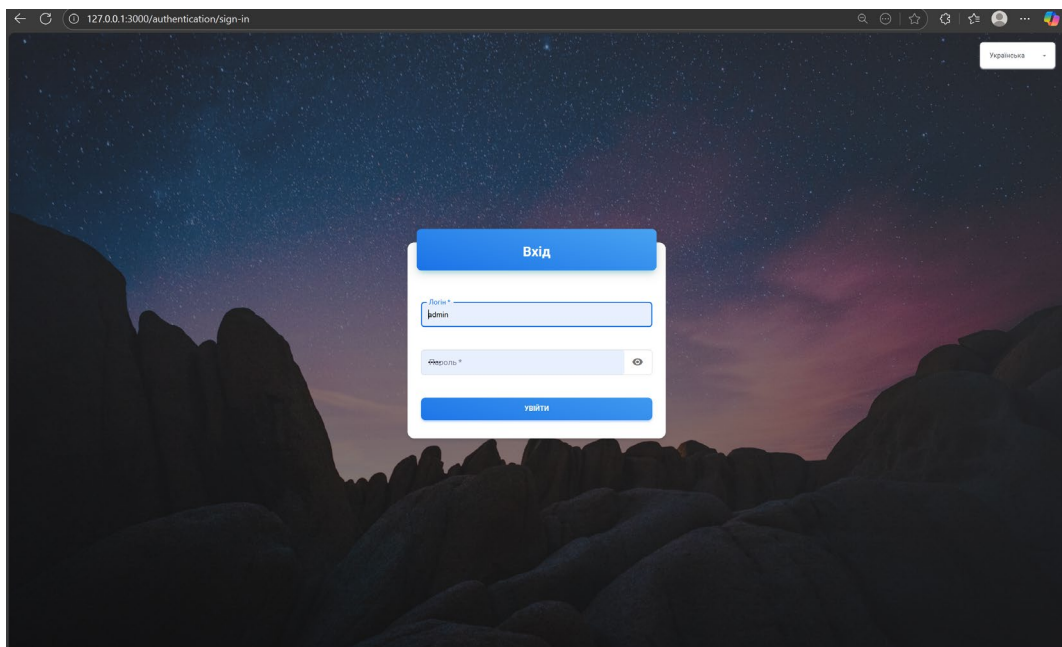


Рисунок 4.1 – Форма входу в систему (Login Screen)

Після успішної авторизації відкривається головна панель керування (Dashboard), яка надає оператору миттєвий зріз стану системи (рис. 4.2).



Рисунок 4.2 – Головна панель керування (Dashboard)

На панелі відображаються ключові показники ефективності (KPI) конвеєрної лінії:

- статус лінії: індикатор "Онлайн/Офлайн";
- продуктивність: швидкість лінії та кількість виробленої продукції;
- статистичні віджети: загальна ефективність (OEE), час простою, кількість активних датчиків та наявність критичних помилок;
- моніторинг у реальному часі: зведені графіки температури, вібрації та тиску.

4.1.2 Адміністрування та налаштування системи

Для гнучкого налаштування системи під конкретні виробничі умови реалізовано модуль адміністрування. Вкладка "Засоби Управління" (рис. 4.3) дозволяє додавати та редагувати канали управління, групувати їх та прив'язувати до конкретних датчиків. Тут задаються діапазони вимірювань, одиниці виміру та класи точності приладів.

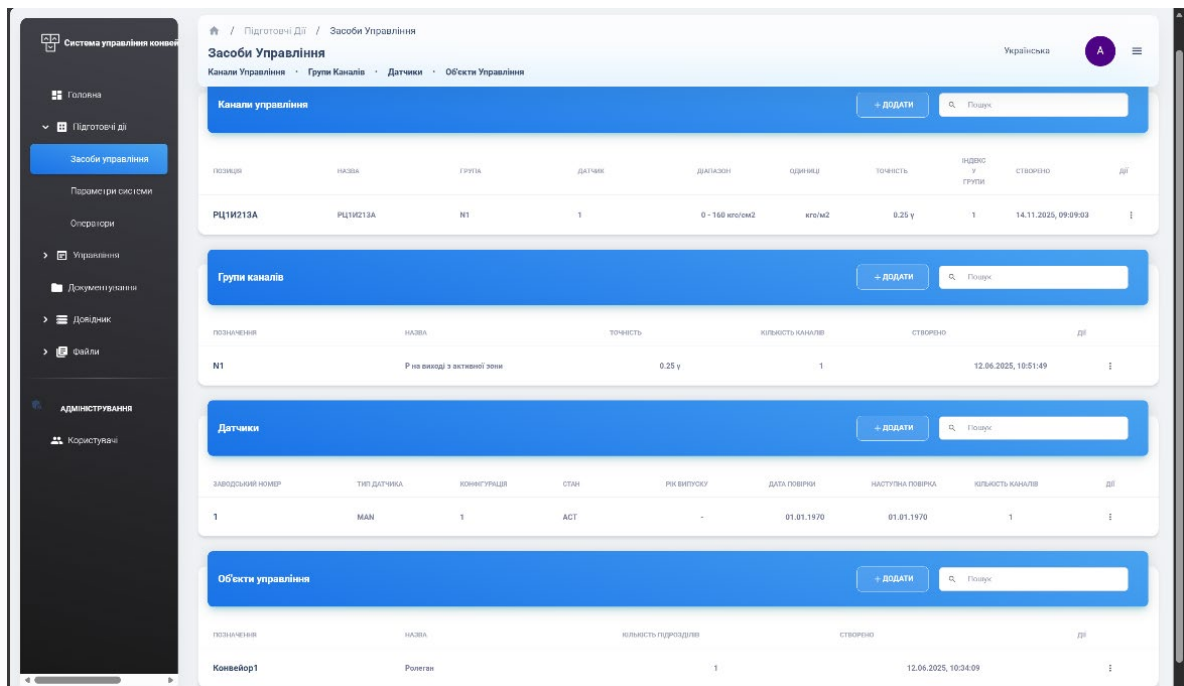


Рисунок 4.3 – Налаштування засобів управління

Модуль "Параметри Системи" (рис. 4.4) дозволяє визначати цикли роботи конвеєра (наприклад, річний цикл експлуатації), налаштовувати часові інтервали моніторингу та визначати режими роботи (Пуск, Зупинка, Стаціонарний режим). Це критично важливо для коректного аналізу дрейфу параметрів.

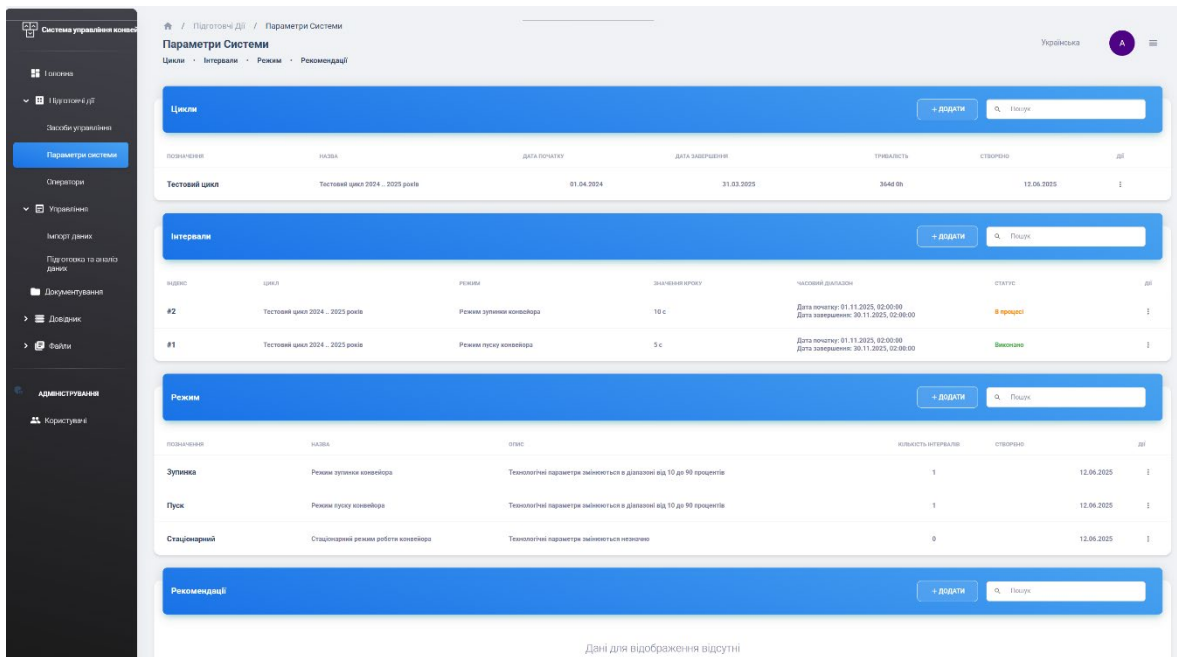


Рисунок 4.4 – Налаштування параметрів системи

Управління персоналом здійснюється через розділ "Оператори" (рис. 4.5), де адміністратор може створювати облікові записи співробітників, призначати їм посади та переглядати історію їхньої діяльності.

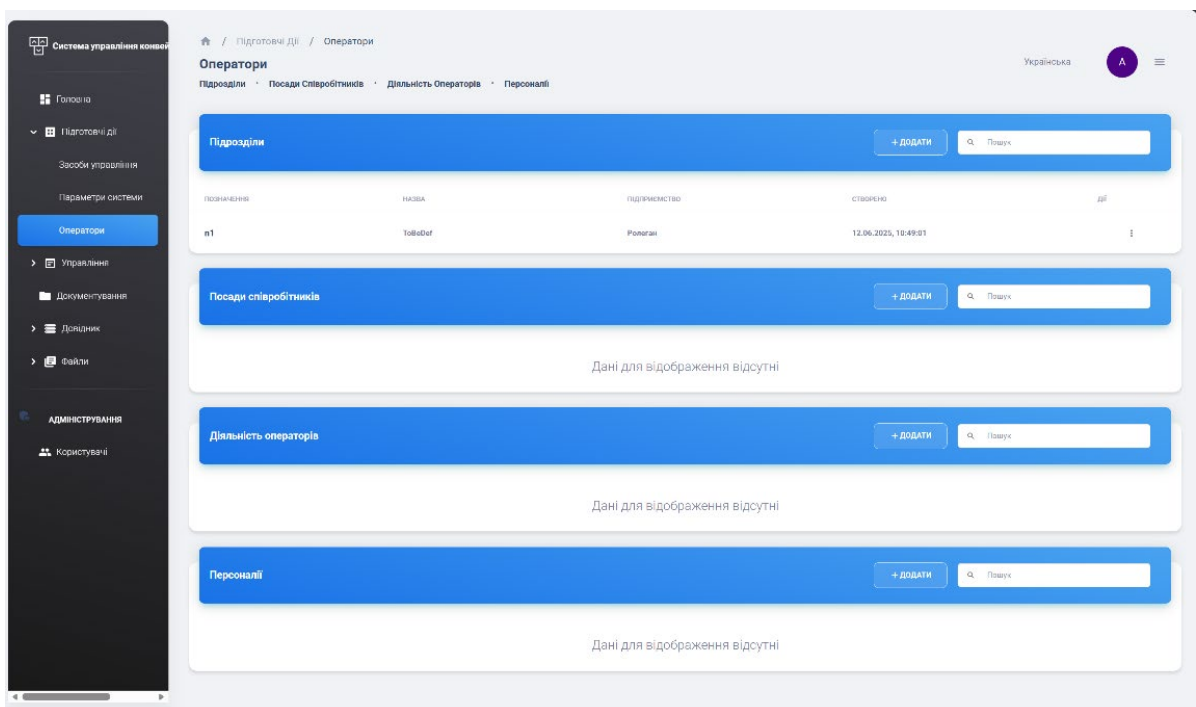


Рисунок 4.5 – Управління операторами

4.1.3 Імпорт та підготовка даних

Ключовою функцією системи є імпорт історичних даних з ПЛК або зовнішніх реєстраторів. Реалізовано інтерфейс імпорту (рис. 4.6), який дозволяє завантажувати файли Excel. Система автоматично парсить файл і надає користувачеві інструмент для вибору діапазонів даних (рис. 4.7), що дозволяє коректно зіставити стовпці файлу з параметрами системи (дати, час, значення датчиків).

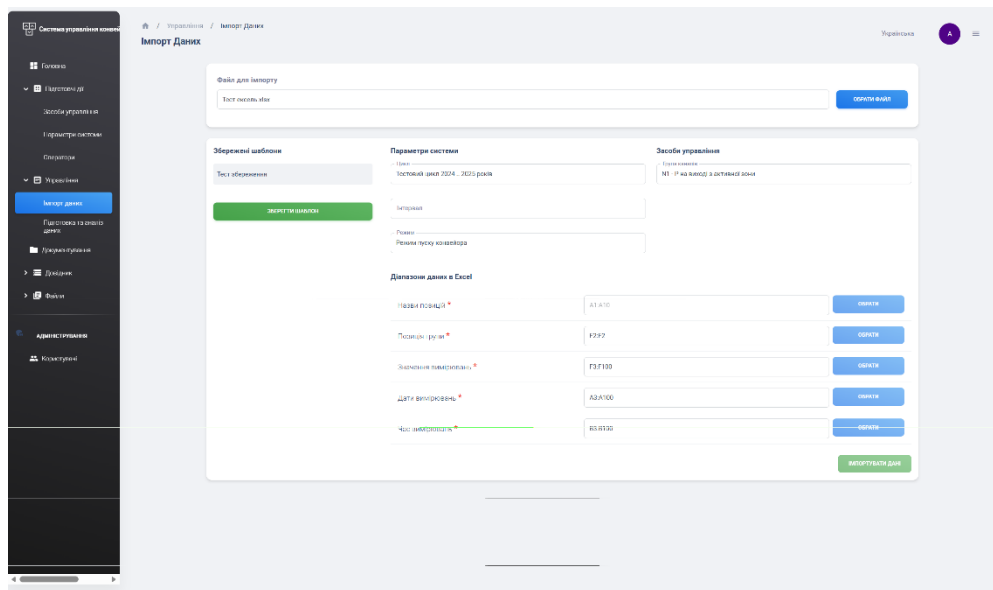


Рисунок 4.6 – Сторінка імпорту даних

Вибір діапазону у завантаженому файлі

	A	B	C	D	E	F	G
1			ФР Різниця тиску ...	ФР Різниця тиску ...	ФР Різниця тиску ...	ФР Різниця тиску ...	ФР Різниця тиску ...
2	Дата	Время	Тест_ВК_01	Тест_ВК_02	Тест_ВК_03	Тест_ВК_04	Тест_ВК_05
3	17.09.2020	00:00:00	4.943	5.005	4.973	4.994	4.818
4	17.09.2020	00:00:10	4.943	5.024	4.982	5.008	4.901
5	17.09.2020	00:00:20	4.943	5.011	4.970	5.006	4.908
6	17.09.2020	00:00:30	4.943	5.009	4.929	5.015	4.858
7	17.09.2020	00:00:40	4.954	5.009	4.935	4.985	4.867
8	17.09.2020	00:00:50	4.935	5.015	5.000	5.017	4.913
9	17.09.2020	00:01:00	4.934	4.992	4.969	4.983	4.861
10	17.09.2020	00:01:10	4.988	5.028	4.992	5.001	4.922
11	17.09.2020	00:01:20	4.949	5.010	4.979	5.001	4.902
12	17.09.2020	00:01:30	4.956	5.012	4.961	5.014	4.841
13	17.09.2020	00:01:40	4.948	5.023	4.951	5.027	4.827
14	17.09.2020	00:01:50	4.969	5.005	4.967	5.005	4.888
15	17.09.2020	00:02:00	4.932	5.025	4.953	5.008	4.911
16	17.09.2020	00:02:10	5.012	5.018	5.028	5.014	4.999
17	17.09.2020	00:02:20	4.986	5.026	4.984	5.027	4.946
18	17.09.2020	00:02:30	5.025	5.029	5.012	4.998	4.976
19	17.09.2020	00:02:40	4.972	5.025	4.944	5.011	4.903
20	17.09.2020	00:02:50	4.998	5.014	4.969	5.021	4.907
21	17.09.2020	00:03:00	4.989	5.020	4.992	5.007	4.921

СКАСУВАТИ ПІДТВЕРДИТИ

Рисунок 4.7 – Вибір діапазону даних у Excel

4.2 Аналіз даних та формування звітності

Після імпорту даних система надає інструменти для їх глибокого аналізу.

4.2.1 Аналіз дрейфів та відхилень

Модуль "Підготовка та Аналіз Даних" дозволяє оператору вибрати конкретний інтервал та переглянути "сирі" значення вимірювань у табличному вигляді (рис. 4.8).

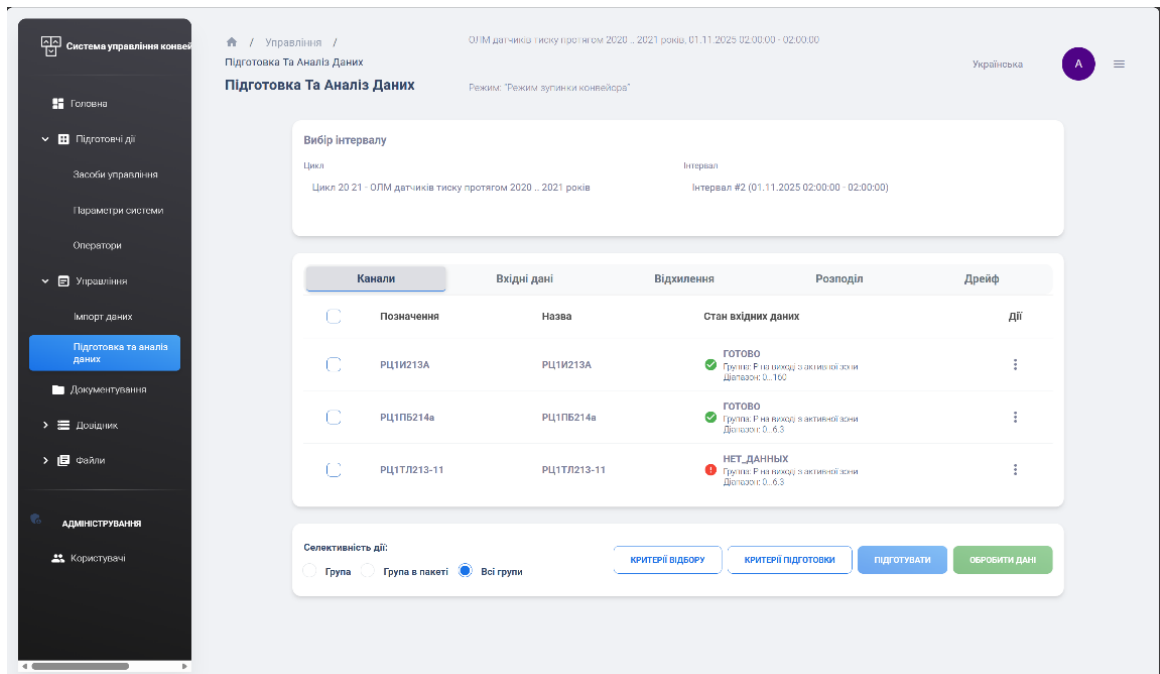


Рисунок 4.8 – Перегляд вхідних даних

На основі цих даних будуються графіки аналізу. Вкладка "Дрейф" (рис. 4.9) візуалізує зміну показань датчиків у часі, дозволяючи виявити повільні відхилення від номіналу, що свідчить про деградацію обладнання. Система автоматично розраховує параметри дрейфу (нижня/верхня межа, ширина коридору) та виводить їх у таблицю під графіком.

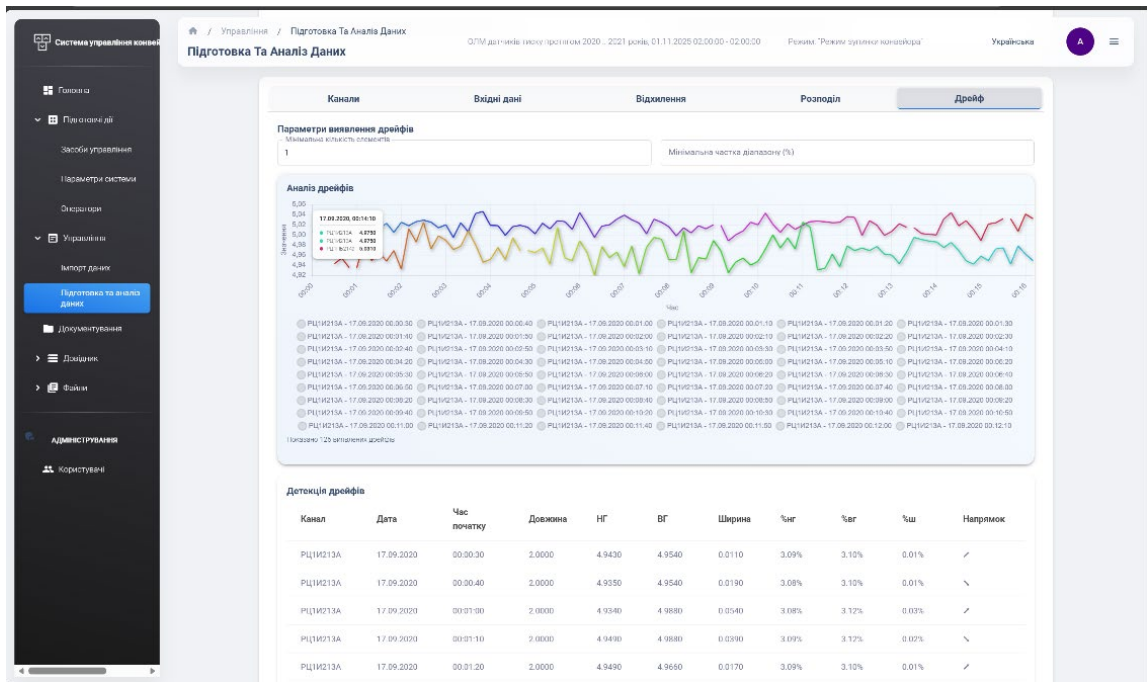


Рисунок 4.9 – Аналіз дрейфів

Вкладка "Розподіл" (рис. 4.10) відображає гістограму розподілу відхилень, що дозволяє оцінити стабільність технологічного процесу та наявність систематичних похибок.

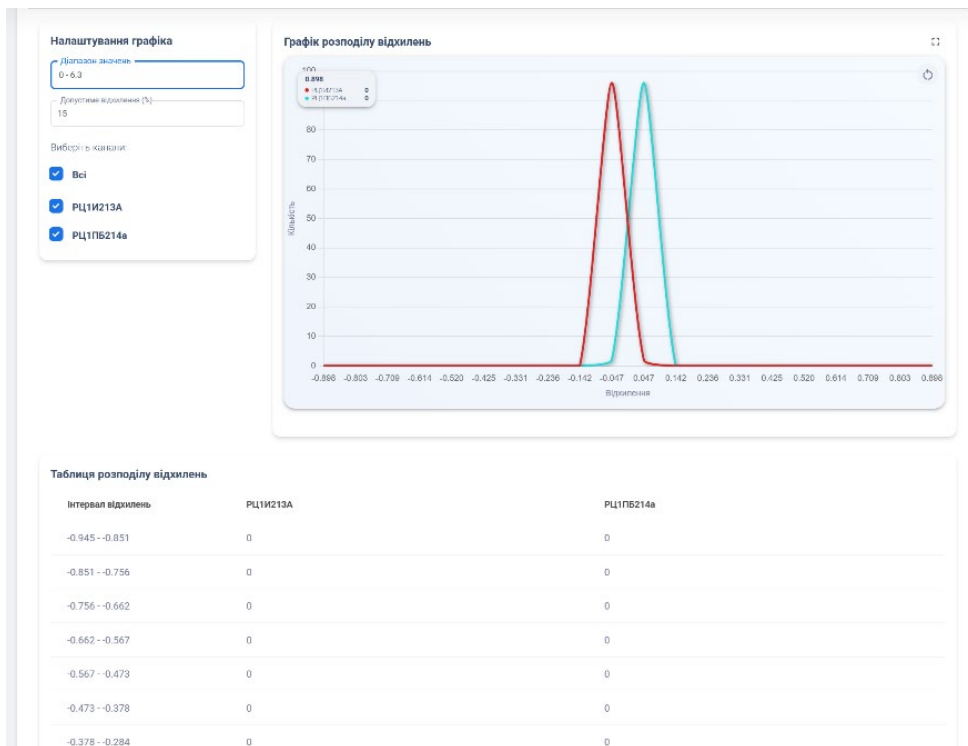


Рисунок 4.10 – Графік розподілу відхилень

4.2.2 Генерація звітної документації

Завершальним етапом роботи є формування офіційної документації згідно з вимогами [1]. Модуль "Документування" дозволяє налаштувати "шапку" звіту, вказати підписантів та перевіряючих (рис. 4.11).

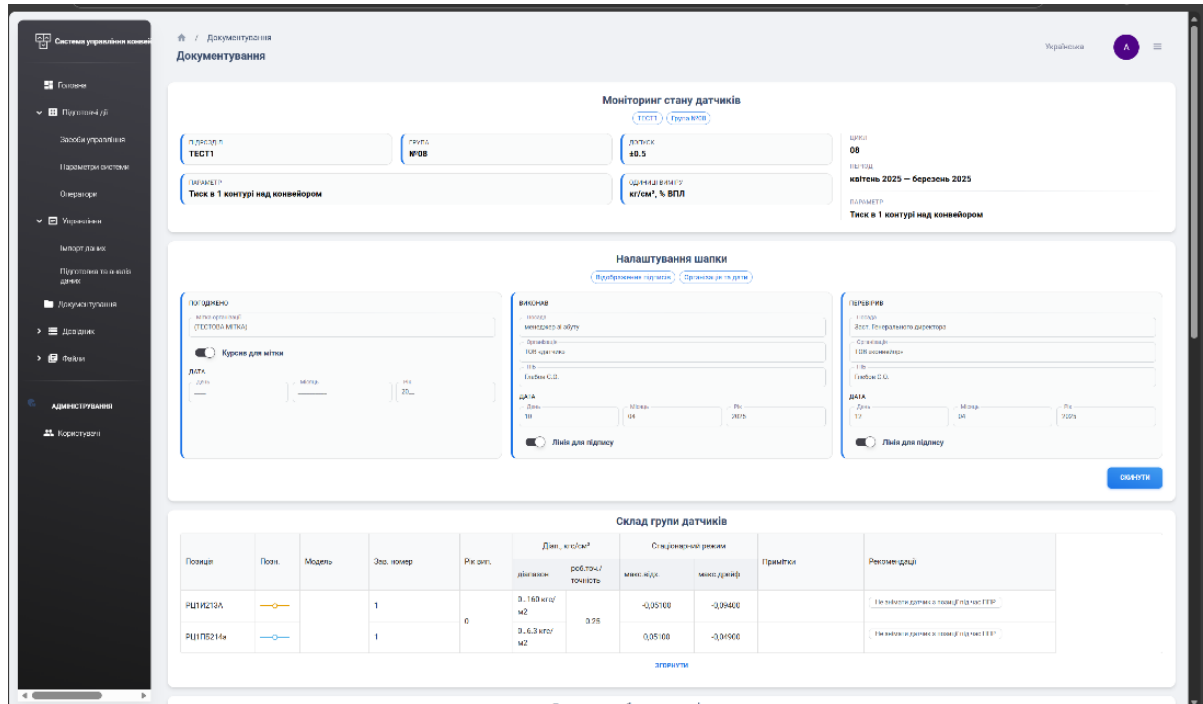


Рисунок 4.11 – Налаштування параметрів звіту

Згенерований документ (рис. 4.12 – 4.13) містить повну інформацію про перевірку: паспортні дані датчиків, графіки перехідних процесів, розраховані значення відхилень та висновки про придатність обладнання до подальшої експлуатації. Звіт може бути експортований у PDF або роздрукований.

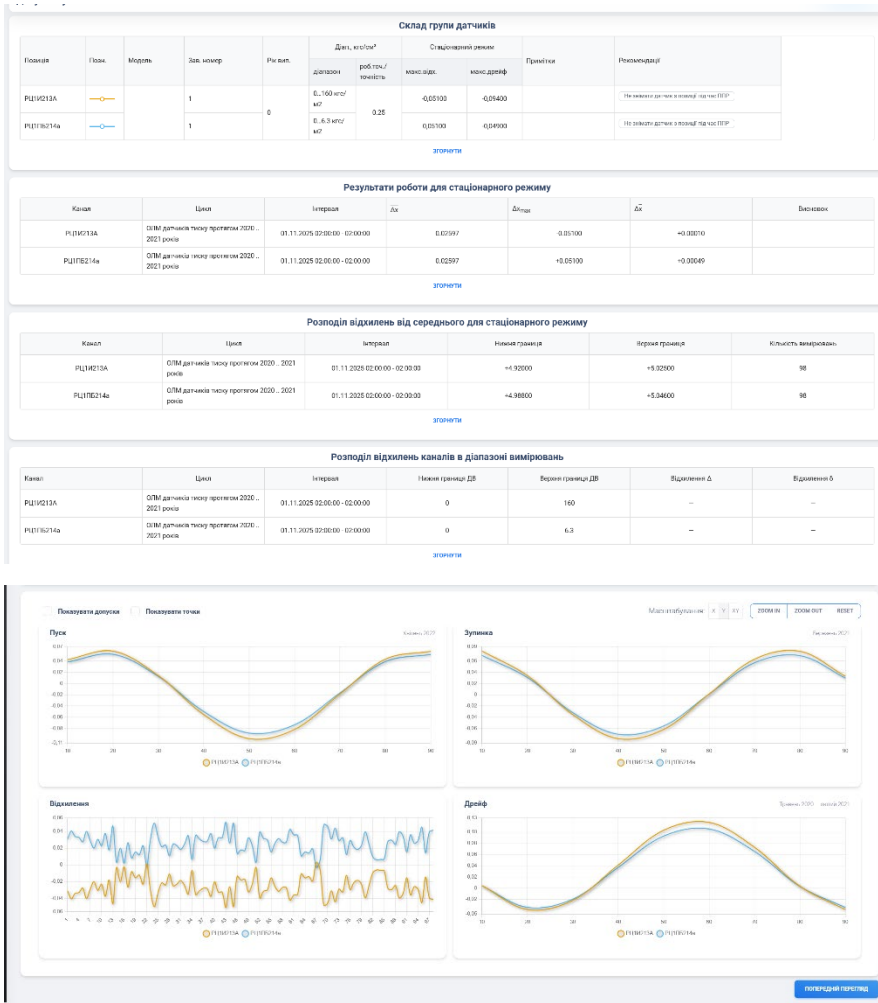


Рисунок 4.12 – Сформований звіт (Частина 1: Таблиці та Графіки)

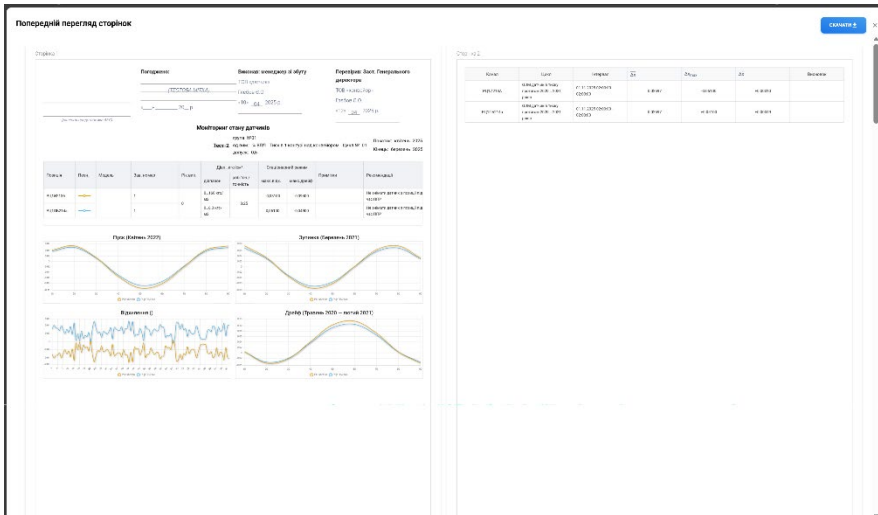


Рисунок 4.13 – Сформований звіт (Частина 2: Таблиці та Графіки)

4.3 Розрахунок надійності апаратного забезпечення системи

Надійність автоматизованої системи управління конвеєрною лінією є комплексною властивістю, яка визначається її здатністю зберігати працездатність протягом заданого часу в умовах експлуатації [17]. Оскільки програмне забезпечення, за умови відсутності критичних помилок, не піддається фізичному зносу, загальна надійність комплексу лімітується надійністю апаратної платформи – робочої станції оператора (АРМ). У цьому підрозділі проведено розрахунок показників надійності для одиночної системи та обґрунтовано необхідність впровадження резервування.

4.3.1 Вихідні дані та методика розрахунку

Для кількісної оцінки надійності використано такі показники:

- інтенсивність відмов (γ): кількість відмов компонента за одиницю часу;
- середній час на відмову (MTBF): математичне сподівання часу роботи до першої відмови, пов'язане з інтенсивністю відмов як $MTBF = \frac{1}{\gamma}$;
- ймовірність безвідмовної роботи (t): ймовірність того, що система не втратить працездатність протягом часу t ;
- коефіцієнт готовності (K_g): ймовірність того, що система буде працездатною у довільний момент часу, враховуючи час на ремонт.

Для розрахунку прийнято модель послідовного з'єднання критичних компонентів, де відмова будь-якого елемента призводить до відмови всієї системи (табл. 4.1). Як базові значення використано MTBF для типових компонентів промислових ПК, що працюють у режимі 24/7.

Таблиця 4.1 – Показники надійності компонентів системи

Компонент системи	MTBF (годин)	Інтенсивність відмов γ , год ⁻⁵	P(t) за 1 рік (8760 год)
Центральний процесор (CPU)	60,000	$1,67 \cdot 10^{-5}$	0,864 (86,4%)
Оперативна пам'ять (RAM)	80,000	$1,25 \cdot 10^{-5}$	0,896 (89,6%)
Твердотільний накопичувач (SSD)	60,000	$1,67 \cdot 10^{-5}$	0,864 (86,4%)
Материнська плата (MB)	30,000	$3,33 \cdot 10^{-5}$	0,749 (74,9%)
Блок живлення (PSU)	25,000	$4,00 \cdot 10^{-5}$	0,705 (70,5%)
Система охолодження (FAN)	60,000	$1,67 \cdot 10^{-5}$	0,864 (86,4%)
Джерело безперебійного живлення (UPS)	20,000	$5,00 \cdot 10^{-5}$	0,645 (64,5%)

4.3.2 Розрахунок надійності одиночної робочої станції

Для одиночної системи без резервування загальна надійність $R_{ПК}$ визначається як добуток ймовірностей безвідмовної роботи всіх її критичних компонентів:

..

$$R_{ПК} = P_{CPU} \cdot P_{RAM} \cdot P_{SSD} \cdot P_{MB} \cdot P_{PSU} \cdot P_{FAN} \cdot P_{UPS},$$

$$R_{ПК} = 0,864 \cdot 0,896 \cdot 0,864 \cdot 0,749 \cdot 0,705 \cdot 0,864 \cdot 0,645 \approx 0,147.$$

Результат: розрахункова ймовірність безвідмовної роботи одиночної станції протягом року без профілактичного обслуговування становить лише 14,7%. Це підтверджує, що експлуатація системи в промислових умовах без заходів щодо підвищення надійності є неприпустимою через високий ризик апаратних збоїв.

4.3.3 Розрахунок коефіцієнта готовності та вплив ремонту

У реальних умовах відмова не означає повну втрату системи, оскільки можливий ремонт. Введемо показник Середнього часу відновлення (MTBF), який для промислового об'єкта (діагностика + заміна вузла) приймемо рівним 8 годин.

Спочатку розрахуємо сумарну інтенсивність відмов системи γ_{total} та її середній час напрацювання на відмову $MTBF_{ПК}$:

$$\gamma_{total} = \sum \gamma_i \approx 1,859 \cdot 10^{-4} \text{ год}^{-1},$$

$$MTBF_{ПК} = \frac{1}{\gamma_{total}} \approx 5378 \text{ годин.}$$

Коефіцієнт готовності K_g для однієї станції:

$$K_g = \frac{MTBF_{ПК}}{MTBF_{ПК} \cdot MTTR} = \frac{5378}{5378 \cdot 8} \approx 0,9985 \text{ (99,85\%).}$$

Хоча коефіцієнт готовності є високим, середній час між відмовами становить лише близько 7 місяців, що може призвести до неприпустимих зупинок виробництва.

4.3.4 Розрахунок надійності системи з гарячим резервуванням

Для підвищення відмовостійкості пропонується використання схеми "гарячого" резервування, що складається з двох ідентичних серверів. Система втрачає працездатність лише у випадку одночасної відмови основного та резервного вузлів.

Коефіцієнт готовності дубльованої системи k_{g_sys} розраховується за формулою:

$$k_{g_sys} = \frac{MTBF_{ПК}^2 + 2 \cdot MTBF_{ПК} \cdot MTTR}{MTBF_{ПК}^2 + 2 \cdot MTBF_{ПК} \cdot MTTR + 2 \cdot MTTR^2}$$

Підставивши значення ($MTBF_{ПК} = 5378, MTTR = 8$):

$$k_{g_sys} = \frac{28922884 + 86048}{28922884 + 86048 + 128} \approx \frac{29008932}{29009060} \approx 0,9999956.$$

Результат: впровадження дублювання підвищує надійність системи до рівня, що перевищує 99,999%. Очікуваний час простою такої системи становитиме лише кілька хвилин на рік.

4.4.5 Вплив планового технічного обслуговування

Навіть для резервованої системи необхідне регулярне технічне обслуговування (ТО) для попередження зносу компонентів (вентиляторів, накопичувачів). Якщо проводити регламентне ТО кожні 3 місяці ($t = 2190$ годин), ймовірність безвідмовної роботи одного ПК за цей період зростає:

$$R_{sys(t_0)} = e^{-1,859 \cdot 10^{-4} \cdot 2189} \approx 0,665 (66,5\%).$$

Надійність дубльованої системи за квартал становитиме:

$$R_{sys(t_0)} = 1 - (1 - 0,665)^2 \approx 1 - 0,112 \approx 0,888 (88,8\%).$$

Висновок: проведений розрахунок доводить, що для забезпечення надійної роботи АСУ КЛ у виробничих умовах використання одиночної робочої станції є неприпустимим. Необхідною умовою є впровадження апаратного дублювання (резервування серверів), використання компонентів промислового класу та проведення регулярного технічного обслуговування не рідше одного разу на квартал.

4.4 Охорона праці та ергономічне забезпечення роботи оператора

Розробка та впровадження автоматизованої системи керування конвеєрною лінією (АСУ КЛ) передбачає тривалу взаємодію оператора з екраном монітора. Тому на етапі проектування було враховано загальні вимоги безпеки та ергономіки, що діють в Україні для організації комп'ютеризованих робочих місць.

Вимоги до організації робочого місця Робоче місце оператора АСУ КЛ організовано таким чином, щоб забезпечити зручне виконання операцій та вільний доступ до органів керування. Просторові параметри робочого місця відповідають чинним санітарним нормам: площа на одне робоче місце становить не менше $6,0 \text{ м}^2$, а об'єм – не менше $20,0 \text{ м}^3$. Конструкція робочого столу дозволяє оптимально розмістити обладнання з урахуванням зони досяжності рук оператора, а робоче крісло має регульовані параметри для підтримки раціональної робочої пози, що дозволяє уникнути статичного навантаження на опорно-руховий апарат.

Вимоги до мікроклімату та освітлення Для підтримання високої працездатності та запобігання перевтомі у приміщенні операторської підтримуються нормовані параметри мікроклімату. Температура повітря забезпечується на рівні $22\text{-}25 \text{ }^\circ\text{C}$, а відносна вологість – у межах $40\text{-}60\%$. Оскільки робота оператора класифікується як зорова робота високої точності, система освітлення спроектована комбінованим методом. Загальна освітленість робочої поверхні забезпечується на рівні $300\text{-}500 \text{ лк}$. Розміщення монітора виконано перпендикулярно до віконних отворів для уникнення відблисків, що знижує втому очей.

Електробезпека Технічні засоби АСУ КЛ живляться від мережі змінного струму напругою 220 В частотою 50 Гц . Згідно з класифікацією приміщень за небезпекою ураження електричним струмом, операторська відноситься до приміщень без підвищеної небезпеки. Для захисту персоналу передбачено комплекс технічних заходів:

- захисне заземлення всіх металевих частин обладнання, що нормально не перебувають під напругою (опір заземлювального пристрою не перевищує 4 Ом);

- використання кабелів з подвійною ізоляцією та їх захист від механічних пошкоджень;

- застосування пристроїв захисного вимкнення (ПЗВ) для запобігання аварійним ситуаціям.

Програмна ергономіка (Інтерфейс) У розробленій системі враховано вимоги інженерної психології до людино-машинних інтерфейсів. Колірна гама мнемокадр обрана у нейтральних тонах, що не викликають швидкої втоми зорового аналізатора. Для відображення критичних подій (аварій, попереджень) використовуються стандартизовані кольори (червоний, жовтий), але без використання надмірних анімаційних ефектів, що можуть відволікати увагу. Шрифтове оформлення забезпечує чітку читабельність текстової інформації з робочої відстані оператора (50-70 см).

Дотримання наведених вимог гарантує безпечні умови праці оператора та мінімізує вплив шкідливих виробничих факторів.

4.6 Висновки до розділу

У четвертому розділі описано практичну реалізацію програмного комплексу АСУ КЛ. Розроблений інтерфейс користувача повністю покриває функціональні вимоги, надаючи інструменти для моніторингу в реальному часі, адміністрування системи та генерації детальних звітів.

Окрему увагу приділено підсистемі роботи з історичними даними. Реалізовано модуль імпорту з Excel, який забезпечує цілісність інформації завдяки багаторівневій валідації та алгоритму контролю якості, що автоматично класифікує дані та фільтрує помилкові значення. Модуль аналізу даних дозволяє візуалізувати дрейф параметрів, що є критичним для предиктивної діагностики.

Також у розділі розглянуто базові питання охорони праці та ергономіки, що підтверджує відповідність розробленого робочого місця оператора загальноприйнятим стандартам безпеки та санітарним нормам.

Проведений розрахунок надійності показав необхідність використання якісних апаратних компонентів та регулярного обслуговування для забезпечення безперебійної роботи системи на виробництві.

ВИСНОВКИ

При виконанні кваліфікаційної роботи було вирішено актуальне науково-прикладне завдання щодо підвищення надійності та оперативності управління конвеєрною лінією шляхом створення спеціалізованого програмного комплексу моніторингу.

У ході дослідження проведено детальний аналіз існуючих методів автоматизації та комунікаційних стандартів, який показав, що перехід на технологію Industrial Ethernet (TCP/IP) дозволяє зменшити затримки передачі даних приблизно у 1500 разів порівняно з традиційним протоколом Modbus RTU, що є критичним для систем реального часу. Також було обґрунтовано доцільність використання стеку веб-технологій (Full-Stack JavaScript) для реалізації HMI, що дозволяє створити гнучке локальне рішення замість дорогих SCADA-систем.

На основі отриманих результатів розроблено архітектуру програмного забезпечення як автономного десктопного застосунку на базі платформи Electron. Запропонована трирівнева структура забезпечує автономність робочого місця оператора та високу швидкість обробки даних. Для ефективного зберігання конфігураційних параметрів та великих масивів історичних даних вимірювань спроектовано реляційну схему бази даних на основі SQLite.

Важливим етапом роботи стала реалізація алгоритмів формалізації та валідації вхідних даних: розроблений модуль імпорту забезпечує коректне зчитування інформації із зовнішніх джерел (Excel) з автоматичною перевіркою цілісності.

Для аналітичної обробки технологічних параметрів створено спеціалізовані програмні модулі, зокрема сервіс CalculatingDataService. Він виконує статистичний аналіз, виявляє «дрейф» показників датчиків та розраховує відхилення від нормальних режимів роботи, що дозволяє

діагностувати стан обладнання на ранніх стадіях.

Взаємодія з оператором здійснюється через розроблений сучасний людино-машинний інтерфейс на базі бібліотеки React, який забезпечує візуалізацію технологічного процесу в реальному часі та автоматичну генерацію звітної документації.

Окрім програмної реалізації, виконано оцінку надійності апаратного забезпечення системи. Розрахунки показали, що для забезпечення безперебійної роботи в промислових умовах із коефіцієнтом готовності понад 99,9% необхідно використовувати схему з гарячим резервуванням серверів.

іТаким чином, мета роботи досягнута: підвищено надійність та оперативність керування конвеєрною лінією шляхом створення спеціалізованого програмного комплексу, що забезпечує аналіз дрейфу технологічних параметрів та діагностику стану обладнання.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. ДСТУ 3008:2015. Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлення. – [Чинний від 2016-07-01]. – Київ : ДП «УкрНДНЦ», 2016. – 26 с.
2. Методичні вказівки з підготовки кваліфікаційної роботи магістра для студентів усіх форм навчання спеціальності 174 “Автоматизація, комп’ютерно-інтегровані технології та робототехніка” освітньої програми “Автоматизація та комп’ютерно-інтегровані технології” / Упоряд.: І.Ш. Невлюдов, А.О. Андрусевич, О.В. Токарєва, С.П. Новоселов, О.В. Сичова. Харків: ХНУРЕ, 2025. – 55с.
3. Глебов Є. О. Переваги протоколу TCP/IP для автоматизованої системи управління конвеєрною лінією із застосуванням засобів людино-машинного інтерфейсу // Global trends in science and education : зб. тез X Міжнар. наук.-практ. конф., м. Київ, 20–22 жовт. 2025 р. – Київ, 2025. – С. 170–173. – URL: <https://sci-conf.com.ua/x-mizhnarodna-naukovo-praktichna-konferentsiya-global-trends-in-science-and-education-20-22-10-2025-kiyiv-ukrayina-arhiv/>.
4. React Documentation [Electronic resource]: Official React Library Documentation. – Meta Platforms, 2024. – Mode of access: <https://react.dev>. – Date of access: 20.11.2025.
5. Node.js Documentation [Electronic resource]: Node.js Reference Documentation. – OpenJS Foundation, 2025. – Mode of access: <https://nodejs.org/docs/latest/api/>. – Date of access: 20.11.2025.
6. Electron Documentation [Electronic resource]: Build cross-platform desktop apps with JavaScript, HTML, and CSS. – OpenJS Foundation, 2024. – Mode of access: <https://www.electronjs.org/docs/latest/>. – Date of access: 20.11.2025.
7. Express.js Routing and Middleware [Electronic resource]: Fast, unopinionated, minimalist web framework for Node.js. – OpenJS Foundation, 2024.

– Mode of access: <https://expressjs.com/en/guide/routing.html>. – Date of access: 20.11.2025.

8. Prisma ORM Documentation [Electronic resource]: Next-generation Node.js and TypeScript ORM. – Prisma Data, Inc., 2024. – Mode of access: <https://www.prisma.io/docs>. – Date of access: 20.11.2025.

9. SQLite Documentation [Electronic resource]: SQL Database Engine. – Hipp, Wyrick & Company, Inc., 2024. – Mode of access: <https://www.sqlite.org/docs.html>. – Date of access: 20.11.2025.

10. Material UI (MUI) Documentation [Electronic resource]: The React UI Library. – MUI, 2024. – Mode of access: <https://mui.com/material-ui/getting-started/>. – Date of access: 20.11.2025.

11. Chart.js Documentation [Electronic resource] : Simple yet flexible JavaScript charting for designers & developers. – Chart.js Contributors, 2024. – Mode of access: <https://www.chartjs.org/docs/latest/>. – Date of access: 20.11.2025.

12. Zustand Documentation [Electronic resource] : A small, fast and scalable bearbones state-management solution. – Poimandres, 2024. – Mode of access: <https://docs.pmnd.rs/zustand/getting-started/introduction>. – Date of access: 20.11.2025.

13. Axios Documentation [Electronic resource] : Promise based HTTP client for the browser and node.js. – Axios, 2024. – Mode of access: <https://axios-http.com/docs/intro>. – Date of access: 20.11.2025.

14. MDN Web Docs: JavaScript [Electronic resource]. – Mozilla Foundation, 2025. – Mode of access: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>. – Date of access: 20.11.2025.

15. Bcrypt.js Documentation [Electronic resource] : Optimized bcrypt in JavaScript with zero dependencies. – 2024. – Mode of access: <https://www.npmjs.com/package/bcryptjs>. – Date of access: 20.11.2025.

16. Le X. A Security-Enhanced Scheme for ModBus TCP Protocol Based on Lightweight Cryptographic Algorithm [Electronic resource] / X. Le, J. Li, Y. Zhao, Z. Fan // Electronics. – 2025. – Vol. 14, iss. 18. – 3674. – Mode of access:

<https://www.mdpi.com/2079-9292/14/18/3674>.

17. NIST Guide to Industrial Control Systems (ICS) Security [Electronic resource] : NIST SP 800-82 Rev. 3. – National Institute of Standards and Technology, 2023. – Mode of access: <https://csrc.nist.gov/pubs/sp/800/82/r3/final>. – Date of access: 20.11.2025.

18. Design and Implementation of Cost Efficient SCADA System for Industrial Automation [Electronic resource] / S. Phuyal, D. Bista, J. Izykowski, R. Bista // International Journal of Engineering and Manufacturing (IJEM). – 2020. – Vol. 10, no. 2. – P. 15–28. – Mode of access: https://www.researchgate.net/publication/340428419_Design_and_Implementation_of_Cost_Efficient_SCADA_System_for_Industrial_Automation.

19. Research on Belt Deviation Fault Detection Technology of Belt Conveyors Based on Machine Vision [Electronic resource] / X. Wu, C. Wang, Z. Tian, X. Huang, Q. Wang // Machines. – 2023. – Vol. 11, iss. 12. – 1039. – Mode of access: <https://www.mdpi.com/2075-1702/11/12/1039>.

20. Web-Based SCADA Systems: A Security Analysis / J. Martins et al. // Future Internet. – 2024. – Vol. 16, Issue 1. – P. 23. – Mode of access: <https://www.mdpi.com/1999-5903/16/1/23>.