

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

ННЦ _____ Заочної форми навчання
(повна назва)

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

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти _____ другий (магістерський)

Розроблення програмного забезпечення для керування мобільною
платформою на базі мережевих технологій у складській логістиці
(тема)

Виконав:
здобувач 2 року навчання,
групи КІТІВЗм-24-1
Владислав ЧЕСНАКОВ
(власне ім'я прізвище)

Спеціальність 174 Автоматизація,
комп'ютерно-інтегровані технології та
робототехніка
(код і повна назва спеціальності)

Тип програми Освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Комп'ютерно-
інтегровані технологічні процеси і
виробництва
(повна назва освітньої програми)

Керівник доцент Роман АРТЮХ
(посада, власне ім'я прізвище)

Допускається до захисту
Зав. кафедри КІТАР

(підпис)

Ігор НЕВЛЮДОВ
(власне ім'я прізвище)

2025 р.

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

«22» грудня 2025 р.



Владислав ЧЕСНАКОВ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

ННЦ	Заочної форми навчання
Кафедра	Комп'ютерно-інтегрованих технологій, автоматизації та робототехніки
Рівень вищої освіти	другий (магістерський)
Спеціальність	174 Автоматизація, комп'ютерно-інтегровані технології та робототехніка
Тип програми	освітньо-професійна
Освітня програма	Комп'ютерно-інтегровані технологічні процеси і виробництва

(код і повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри КІТАР _____
(підпис)

«___» _____ 2025 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві _____ *Чеснакову Владиславу Сергійовичу*
(прізвище, ім'я, по батькові)

- Тема роботи *Розроблення програмного забезпечення для керування мобільною платформою на базі мережесих технологій у складській логістиці*
затверджена наказом по університету від "03" листопада 2025 р. № 192 Стз.
- Термін подання студентом роботи "24" грудня 2025 р.
- Вихідні дані до роботи 3.1 Мобільна платформа;
3.2 Симуляційне середовище Webots;
3.3 Мова програмування C++, Python;
3.4 MQTT-протокол;
3.5 Програмні та апаратні обмеження, пов'язані з реальним часом роботи системи.
- Перелік питань, що потрібно опрацювати в роботі 4.1 Вступ;
4.2 Аналіз сучасних технологій у складській логістиці;
4.3 Розроблення програмного забезпечення;
4.4 Техніко-економічне обґрунтування впровадження програмного забезпечення;
4.5 Забезпечення безпечних умов праці;
4.6 Висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій *Демонстраційний матеріал представлений у форматі презентації PowerPoint (*.ppt) – 14 с. формату А4*

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

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

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	<i>Аналіз сучасних технологій у складській логістиці</i>	03.11 – 10.11.25	<i>виконано</i>
2	<i>Розроблення програмного забезпечення</i>	11.11 – 03.12.25	<i>виконано</i>
3	<i>Техніко-економічне обґрунтування впровадження програмного забезпечення</i>	03.12 – 06.12.25	<i>виконано</i>
4	<i>Охорона праці та безпека життєдіяльності</i>	06.12 – 08.12.25	<i>виконано</i>
5	<i>Оформлення пояснювальної записки</i>	09.12 – 17.12.25	<i>виконано</i>
6	<i>Подання роботи на перевірку Інтернет-системою StrikePlagiarism</i>	17.12.25	<i>виконано</i>
7	<i>Подання роботи на рецензію</i>	18.12.25	<i>виконано</i>
8	<i>Подання роботи на підпис зав. кафедри</i>	19.12 – 23.12.25	<i>виконано</i>
9	<i>Подання кваліфікаційної роботи в ЕК</i>	24.12.25	<i>виконано</i>

Дата видачі завдання 03.11.2025 р.

Здобувач _____
(підпис)

Владислав ЧЕСНАКОВ

Керівник роботи _____
(підпис)

доцент Роман АРТЮХ
(посада, власне ім'я прізвище)

РЕФЕРАТ

Пояснювальна записка: 73 с., 3 табл., 9 рис., 2 дод., 18 джерел.

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, МОБІЛЬНА ПЛАТФОРМА, РОБОТИЗОВАНА СИСТЕМА, СКЛАДСЬКА ЛОГІСТИКА, МЕРЕЖЕВІ ТЕХНОЛОГІЇ, КЛІЄНТ-СЕРВЕРНА АРХІТЕКТУРА, АВТОМАТИЗАЦІЯ.

Мета роботи – розроблення програмного забезпечення для керування мобільною платформою на базі мережових технологій з метою підвищення ефективності складських логістичних процесів.

Об'єкт дослідження – процес керування мобільними роботизованими платформами у складській логістиці.

Предмет дослідження – програмне забезпечення та мережові технології, що використовуються для керування мобільною платформою в складському середовищі.

Методи дослідження – методи об'єктно-орієнтованого програмування, моделювання, використання клієнт–серверної архітектури та мережових протоколів обміну даними, експериментальне тестування у симуляційному середовищі.

Результатом цієї роботи є розроблене програмне забезпечення для керування мобільною платформою, яке забезпечує ручний та автономний режими руху, уникнення перешкод і контроль меж робочої зони з використанням мережової взаємодії. Реалізована система дозволяє організувати ефективний обмін даними між серверною частиною та контролером платформи.

Отримані результати можуть бути використані як основа для створення реальних автоматизованих систем керування мобільними платформами у складській логістиці.

ABSTRACT

Explanatory note: 73 pp., 3 tab., 9 fig., 2 appendices, 18 sources.

SOFTWARE, MOBILE PLATFORM, ROBOTIC SYSTEM, WAREHOUSE LOGISTICS, NETWORK TECHNOLOGIES, CLIENT–SERVER ARCHITECTURE, AUTOMATION.

The purpose of the work is to develop software for controlling a mobile platform based on network technologies in order to improve the efficiency of warehouse logistics processes.

The object of the research is the process of controlling mobile robotic platforms in warehouse logistics.

The subject of the research is software and network technologies used for controlling a mobile platform in a warehouse environment.

Research methods include analysis of object-oriented programming methods, modeling, the use of client-server architecture and network communication protocols, as well as experimental testing in a simulation environment.

The result of this work is the developed software for controlling a mobile platform that provides manual and autonomous motion modes, obstacle avoidance, and workspace boundary control using network interaction. The implemented system enables efficient data exchange between the server-side component and the platform controller.

The obtained results can be used as a basis for developing real automated control systems for mobile platforms in warehouse logistics.

ЗМІСТ

Перелік умовних скорочень	9
Вступ	10
1 Аналіз сучасних технологій у складській логістиці	12
1.1 Роль автоматизації та цифровізації	12
1.2 Роль автоматизації та цифровізації	16
1.3 Використання мобільних платформ і роботів у складах	20
1.4 Огляд мережевих технологій	26
1.5 Аналіз існуючих програмних рішень	30
1.6 Постановка задачі на розробку програмного забезпечення	32
1.7 Висновки до розділу	34
2 Розроблення програмного забезпечення	36
2.1 Вибір мов програмування та програмних інструментів	36
2.2 Проектування архітектури програмної системи	39
2.3 Розробка комунікаційного протоколу та структури інформаційних повідомлень	42
2.4 Реалізація контролера мобільної платформи	43
2.5 Реалізація серверної частини системи управління логістичними місіями	46
2.6 Імітація роботи мобільної	49
2.7 Тестування та аналіз результатів роботи системи	52
2.8 Висновки до розділу	53
3 Техніко-економічне обґрунтування впровадження програмного забезпечення	61
3.1 Обґрунтування доцільності автоматизації складських операцій	61
3.2 Вихідні дані для економічних розрахунків	62
3.3 Розрахунок витрат на ручне виконання складських операцій	62

3.4 Розрахунок витрат на автоматизований варіант	63
3.5 Розрахунок економічного ефекту від впровадження	63
3.6 Розрахунок терміну окупності системи	64
3.7 Додаткові переваги впровадження системи	65
3.8 Висновки до розділу	65
4 Охорона праці та безпека життєдіяльності	66
4.1 Загальна характеристика умов праці та нормативно-правова база	66
4.2 Аналіз небезпечних і шкідливих виробничих факторів	67
4.3 Організаційні та технічні заходи забезпечення безпеки	68
4.4 Вимоги електробезпеки та пожежної безпеки	68
4.5 Ергономічні та санітарно-гігієнічні вимоги	69
4.6 Дії персоналу в аварійних ситуаціях та висновки до розділу	69
Висновки	70
Перелік джерел посилання	72
Додаток А Демонстраційний матеріал	74
Додаток Б Програмний код серверу	75

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

AGV – Automated Guided Vehicles;

AMR – Autonomous Mobile Robots;

AS/RS – Automated Storage and Retrieval Systems;

FMS – Fleet Management System;

IoT – Інтернет речей;

ПЗ – програмне забезпечення;

RMC – Robot Management Console;

SLAM – Simultaneous Localization and Mapping;

WES – Warehouse Execution System;

WMS – Warehouse Management System.

ВСТУП

Нинішній розвиток складської логістики характеризується зростанням обсягів вантажообігу, ускладненням внутрішньоскладських процесів та підвищенням вимог до швидкості, точності й безпеки виконання логістичних операцій. В умовах цифровізації промисловості та впровадження концепцій Industry 4.0 особливої актуальності набуває автоматизація внутрішньоскладського транспорту, яка дозволяє підвищити ефективність використання ресурсів, зменшити витрати та мінімізувати вплив людського фактора.

Одним із перспективних напрямів автоматизації складських процесів є використання мобільних роботизованих платформ, які здатні автономно або напіваавтономно виконувати транспортні та допоміжні операції. Ефективність таких систем значною мірою залежить від програмного забезпечення керування, яке забезпечує взаємодію між мобільною платформою, оператором і інформаційною інфраструктурою складу. Велику роль у даному контексті відіграють мережеві технології, які дозволяють реалізувати гнучку клієнт-серверну архітектуру керування, забезпечити масштабованість системи та її інтеграцію з іншими інформаційними системами підприємства.

Розроблення власного програмного забезпечення є актуальним, незважаючи на наявність великої кількості комерційних рішень у сфері складської автоматизації. Це пов'язано з необхідністю адаптації систем до конкретних умов експлуатації, оптимізації витрат, а також можливістю розширення функціональних можливостей відповідно до потреб підприємства. Крім того, використання відкритих програмних технологій і стандартних мережевих протоколів дозволяє створювати гнучкі та економічно доцільні рішення.

У зв'язку з цим має актуальність задача розроблення програмного забезпечення для керування мобільною платформою на базі мережевих

технологій у складській логістиці, яке дозволило забезпечити б ефективне, надійне та безпечне виконання транспортних операцій у складському середовищі.

Об'єкт дослідження – процес управління мобільною платформою у складській логістиці.

Предмет дослідження – програмне забезпечення та мережеві технології, що використовуються для керування мобільною платформою в складському середовищі.

Мета роботи – розробка програмного забезпечення для керування мобільною платформою на базі мережевих технологій у складській логістиці.

Для досягнення поставленої мети планується виконати наступні завдання:

- проаналізувати сучасні підходи до автоматизації складської логістики та використання мобільних роботизованих платформ;
- дослідити існуючі методи та архітектури мережевого керування мобільними платформами;
- сформулювати вимоги до програмного забезпечення керування мобільною платформою;
- розробити архітектуру програмної системи керування на основі клієнт–серверної моделі;
- реалізувати програмне забезпечення з використанням мережевих технологій;
- провести тестування розробленої системи в симуляційному середовищі;
- оцінити техніко-економічну доцільність впровадження системи;
- розглянути питання охорони праці та безпеки життєдіяльності при експлуатації мобільної платформи;
- оформити кваліфікаційну роботу згідно ДСТУ 3008:2015 [1], а також за допомогою методичних вказівок [2-3],.

1 АНАЛІЗ СУЧАСНИХ ТЕХНОЛОГІЙ У СКЛАДСЬКІЙ ЛОГІСТИЦІ

1.1 Роль автоматизації та цифровізації

Розвиток технологій Індустрії 4.0 змушує сучасну логістику переживати глибоку трансформацію. Традиційні методи обліку такі, як планування та керування запасами більше не здатні забезпечити необхідну швидкість, точність і гнучкість у мінливому ринковому середовищі. У відповідь на це формується нова парадигма – «Логістика 4.0», яка ґрунтується на використанні таких технологій, як кіберфізичні системи (CPS), Інтернет речей (IoT), Big Data та штучний інтелект (AI) [6].

Основною метою даної концепції, є створення інтелектуального саморегульованого логістичного середовища, у якому всі об'єкти (транспорт, товари, складське обладнання, персонал) об'єдані у єдину цифрову екосистему.

Перехід від класичної логістики до цифрової ведеться кардинальними змінами у структурі управління складом. Якщо раніше центром системи являлась людина, яка здійснювала прийняття рішень на основі досвіду, то нині центром виступають дані, а рішення приймаються автоматично системами керування, які аналізують потоки інформації у реальному часі.

Цифровізація – це основа концепції Логістики 4.0. Її сутність полягає у переведенні всіх операцій та інформаційних потоків у цифрову форму. У традиційній логістиці не аби яка частина даних велася на паперових носіях або вводилася вручну, що створювало ризики помилок, затримок та втрати інформації. У цифровій системі всі дії працівників, рухи товарів і техніки фіксуються в автоматичному режимі.

Основними інструментами цифровізації є:

– WMS (Warehouse Management System) – система управління складом, яка є центром (мозком) логістичних операцій. Вона здійснює координацію всіх процесів: приймання, зберігання, комплектації, та відвантаження. WMS зберігає точні координати розміщення кожної одиниці товару, розподіляє завдання між персоналом і технікою, оптимізує маршрути переміщень[7];

– технології збору даних – замість ручного введення використовуються, такі технології як штрих-коди, радіочастотна ідентифікація та системи комп'ютерного зору, які автоматично розпізнають об'єкти та контролюють правильність операцій;

– Big Data та аналітика – накопичення великих обсягів даних дозволяє створювати прогностичні моделі попиту, оптимізувати маршрути, визначати «вузькі місця» у роботі складу.

Результатами цифровізації, є прозорість, яка дає можливість у будь-який момент відстежити розташування товару, стан запасів та виконання завдань. Точність даних, які мінімізують людський фактор, тобто зменшують кількість помилок. Аналітичні можливості – система має здатність не лише фіксувати, але й аналізувати події. А також створення цифрового двійника – віртуальної моделі складу, яка відображає усі фізичні процеси в реальному часі. Цифровий двійник дозволяє тестувати нові схеми розміщення товарів, оптимізувати маршрути роботів чи навантажувачів, прогнозувати ефективність змін без ризику збоїв у реальному середовищі.

Таким чином, цифровізація створює базу для наступного етапу – автоматизації, де отримані дані стають підґрунтям для виконання фізичних дій без участі людини.

Автоматизація – це використання технічних засобів і систем керування для виконання завдань, які раніше виконувалися вручну. Її впровадження дозволяє в разі підвищити продуктивність, зменшити кількість помилок і скоротити витрати.

Основними прикладами автоматизації можна назвати[8]:

- конвеєрні системи, які забезпечують безперервне транспортування товарів між зонами складу;
- автоматизовані сортувальні лінії, які дозволяють ідентифікувати товар за штрих-кодом чи RFID-міткою та направити його у потрібну зону;
- крани-штабелери (AS/RS – Automated Storage and Retrieval Systems) – механізми, які автоматично розміщують і дістають товари з полиць на основі команд від WMS;
- автоматичні пакувальні машини, які здійснюють контроль ваги, маркування та якість упаковки.

Серед переваг автоматизації можна виділити наступне:

- підвищення пропускної здатності складу;
- зменшення часу на обробку замовлень;
- зниження впливу людського фактора;
- економія на енергоресурсах та оптимізація простору.

Отже, автоматизацію можна назвати ключем для стабільної роботи логістичного центру. Проте її ефективність без цифровізації буде обмеженою, адже без точних даних система не зможе приймати коректні рішення.

Роботизація виступає, як наступний етап автоматизації. Це більш гнучкий та розумний етап автоматизації. Якщо автоматизоване обладнання виконує строго задані дії (наприклад, рух по фіксованій лінії), то робот здатен адаптуватися до змін середовища, приймати рішення на основі даних і навіть взаємодіяти з людьми чи іншими машинами.

Можна навести наступні приклади сучасної роботизації:

- AGV (Automated Guided Vehicles) – автоматичні транспортні візки, які рухаються по маркованих маршрутах [9];
- AMR (Autonomous Mobile Robots) – автономні мобільні роботи, що орієнтуються у просторі за допомогою сенсорів і карт приміщення [9];

- роботизовані маніпулятори – використовуються для комплектації замовлень (picking) або завантаження/розвантаження товарів;

- дрони – у великих складах застосовуються для інвентаризації, сканування штрих-кодів на верхніх стелажах або контролю безпеки.

Використання роботів дозволяє досягти високої продуктивності при мінімальній кількості персоналу. Роботизація також підвищує безпеку праці, оскільки машини беруть на себе небезпечні або рутинні завдання.

Цифровізація, автоматизація та роботизація – це три взаємопов'язані елементи єдиної системи.

Без цифрових даних робот не може знати, куди їхати; без автоматизованих механізмів дані залишаються лише інформацією; без роботизації система не має гнучкості й автономності.

Типовий приклад взаємопов'язаності вищевказаних елементів системи:

- WMS (цифровізація) аналізує дані про замовлення;
- вона формує завдання для мобільного робота (роботизація);
- робот отримує команду привезти товар зі стелажа, який обслуговує автоматизована система (AS/RS);
- усі дії відбуваються без втручання людини, але під її контролем через цифровий інтерфейс.

Отже можна зробити висновок до даного підпункту автоматизація та цифровізація – це не просто технологічний тренд, а стратегічна необхідність сучасної логістики. Вони забезпечують:

- ефективність та зниження витрат;
- гнучкість системи у відповідь на зміну попиту;
- прозорість і керованість усіх процесів;
- передумови для впровадження штучного інтелекту та повної автономізації складу.

У результаті підприємства переходять від моделі «реагування» до моделі «передбачення і саморегуляції», коли система здатна самостійно прогнозувати навантаження, оптимізувати маршрути й адаптуватися до зовнішніх умов.

Саме на цьому перетині – цифровізації, автоматизації та роботизації – формується нова епоха логістичних систем, до якої належить і досліджувана у цій роботі тема – розробка програмного забезпечення (ПЗ) для керування мобільною платформою у складській логістиці.

1.2 Використання мобільних платформ і роботів у складах

Останніми десятиліттями логістична галузь зазнає радикальних змін під впливом автоматизації та роботизації. Якщо раніше основною складовою роботи складу були люди, які виконували всі фізичні операції – від транспортування до комплектації товарів – то нині все частіше ці завдання виконують мобільні платформи та роботи.

Такі системи стають ключовим елементом концепції розумного складу (Smart Warehouse), де всі процеси – від приймання до їх відвантаження – відбуваються в автоматичному або напівавтоматичному режимі під управлінням централізованої WMS-системи.

Використання мобільних роботів дозволяє суттєво знизити витрати на персонал, мінімізувати кількість помилок і прискорити обробку замовлень. За даними DHL Robotics Report 2023, автоматизовані склади з мобільними платформами обробляють замовлення в середньому на 30 – 40 % швидше, ніж звичайні де працюють в основному люди.

Ключовим елементом таких систем є мобільні платформи – автономні або напівавтономні пристрої, призначені для транспортування вантажів усередині складу. Основна класифікація таких пристроїв ґрунтується на рівні їхньої автономності – AGV (Automated Guided Vehicles) та AMR (Autonomous Mobile Robots).

Отже перейдемо до розбору вищезазначених пристроїв, AGV –

транспортні пристрої, які позначаються магнітними стрічками, оптичними лініями або лазерними відбивачами. Такі транспортні засоби отримують командивід центрального контролера, який задає маршрут, зупинки та швидкість руху.

Дана система, є «жорстко автоматизованою». Якщо на її шляху з'являється перешкода (наприклад, людина або якийсь інший предмет), AGV просто зупиняється і чекає доки шлях не буде звільнено.

Попри обмежену гнучкість, AGV також має низку переваг, а саме:

- висока надійність – рух завжди передбачений, а маршрути стабільні;
- простота обслуговування – система не потребує складного програмування;
- довгий термін служби при мінімальних оновленнях ПЗ.

Але, через жорстку прив'язку до маршруту AGV не підходять для динамічних середовищ, де маршрути можуть змінюватися або де постійно переміщуються люди.

AGV і досі використовуються на великих промислових підприємствах, де необхідно регулярно здійснювати перевезення вантажів за постійними маршрутами. Наприклад:

- транспортування палет між зонами виробництва та зберігання;
- доставка матеріалів на конвеєр;
- робота у холодильних складах, де наявні важкі умови для персоналу.

Отже, AGV залишається популярною технологією завдяки своїй простоті, передбачуваності та надійності, але із-за стрімкого розвитку технологій поступово витісняється більш гнучкими системами – AMR

AMR – є новим поколінням мобільних платформ, здатних самостійно орієнтуватися у просторі, будувати карту середовища та планувати власний маршрут. Їх зазвичай називають «розумними роботами складу».

Основною відмінністю AMR системи від AGV – інтелектуальна навігаціяю Такі роботи використовують технології:

- SLAM (Simultaneous Localization and Mapping) – технологія, що

здійснює забезпечення функціонування мобільних без потреби у стаціонарному штативі. Технологія дозволяє динамічно відстежувати положення пристрою в реальному часі, забезпечуючи точний і високоякісний збір просторових даних, навіть у складних масштабних середовищах. Простіше кажучи, здійснює одночасну локалізацію та побудову карти [10];

- Lidar (лазерний далекомір) – дана технологія отримує та обробляє інформацію про віддалені об'єкти за допомогою активних оптичних систем, що використовують явища поглинання і розсіювання світла в оптично прозорих середовищах. Тобто, здійснює сканування простору у 360° для виявлення перешкод [11];

- 3D-камери та ультразвукові сенсори – завдяки ним здійснюється розпізнавання об'єктів, людей, полиць;

- AI-модулі використовуються для адаптивного прийняття рішень у складних ситуаціях.

Вищевказані технології дозволяють AMR не лише здійснювати самостійне пересування, але й оптимізувати маршрут у режимі реального часу. Якщо робот зустрічає перешкоду, він не зупиняється, а перебудовує маршрут та продовжує рух у задану точку.

Впровадження AMR забезпечить широкий спектр переваг, які виходять за рамки простої автоматизації. Ці інтелектуальні системи змінюють те, як об'єкти керують внутрішньою логістикою, підвищують безпеку та адаптуються до змінних операційних потреб. AMR забезпечує вимірні переваги в продуктивність, гнучкість та зниження витрат [12].

AMR стали стандартом для гнучкої логістики, особливо у великих розподільчих центрах, таких як Amazon, DHL, Ocado. Вони виконують широкий спектр завдань:

- перевезення контейнерів між зонами;
- доставку стелажів до працівників;
- участь у комплектації замовлень;
- автоматичну інвентарізацію.

Сучасні AMR часто працюють у «змішаних середовищах», де поруч з ними діють люди та інша техніка. Це дозволяє створювати гібридні системи, у яких зберігається людський контроль при максимальній автоматизації процесів.

Розберемо далі типи та функції мобільних платформ у складі, велику кількість появи яких було зумовлено різноматніми задачами у логістичних центрах. Нижче наведено основні категорії:

- сортувальні роботи – компактні AMR, які здійснюють транспортування окремих посилок до відповідних сортувальних лотків або зон. Вони часто працюють у великих кількостях, створюючи рої роботів, що взаємодіють між собою. Завдяки централізованому алгоритму управління система може здійснювати обробку тисячі посилок на годину, що робить такі роботи незамінними у поштових та e-commerce центрах;

- роботи типу «товар-до-людини» – одна із найефективніших концепцій у сучасній логістиці. Робот під'їжджає під цілий стелаж із товарами і транспортує його до робочого місця комплектувальника. Людина не ходить по складу, а товари доставляються до неї;

- автономні навантажувачі – безпілотні аналоги традиційних навантажувачів. Вони оснащені лидарами, камерами та системами виявлення палет. Такі машини здатні самостійно завантажувати і розвантажувати вантажівки, розміщувати товари на високих стелажах. Використовуються у великих логістичних центрах і на підприємствах із високим товарообігом, де безпека та точність операцій мають критичне значення;

- AMR-тягачі, які здатні буксирувати кілька візків одночасно, доставляючи вантажі по внутрішніх маршрутах складу або між зонами. Їх часто застосовують у виробничих процесах, наприклад у автомобільній промисловості, де потрібно безперервно здійснювати підвоз деталей до конвеєрів.

Мобільні платформи та роботи діють не автономно, а у взаємодії з WMS або системами класу MES/ERP.

Центральна системи керування забезпечує виконання таких функцій:

- призначає завдання конкретному роботу;
- оптимізує маршрути з урахуванням поточного завантаження складу;
- координує кількох роботів одночасно, запобігаючи зіткненням;
- відстежує стан батарей і планує зарядку.

Ця взаємодія дозволяє сформувати єдиний «цифровий організм» складу, де всі елементи – від сенсорів до роботів – обмінюються даними у режимі реального часу.

Використання мобільних платформ і роботів у складах є центральним напрямом сучасної логістики. Від AGV, що працюють за жорсткими маршрутами, до AMR, здатних мислити і адаптуватися – розвиток робототехніки докорінно змінює підхід до організації складських процесів.

Мобільні роботи дозволяють:

- підвищити продуктивність у 2-3 рази;
- скоротити витрати на персонал;
- забезпечити безперервну роботу 24/7;
- підвищити безпеку праці та точність обліку.

Сьогодні AMR стали не просто інструментом, а невід'ємною частиною цифрової екосистеми складу, яка об'єднує автоматизацію, роботизацію та аналітику в єдину інтегровану систему.

1.3 Огляд мережевих технологій

У сучасних логістичних системах інформація є таким самим важливим, як і товари або транспорт. Вона забезпечує зв'язок між усіма елементами «розумного складу» – від WMS до окремих роботів, сенсорів та операторів. Щоб парк мобільних платформ працював у взаємодії, потрібна високошвидкісна, надійна та безшовна мережева інфраструктура. Вона є «нервовою системою» складу, через яку передаються команди, телеметрія, відеопотоки з камер, дані з Lidar-сенсорів та результати обчислень з периферійних вузлів.

З огляду на це, вибір правильної мережевої технології є критичним етапом при проектуванні сучасних логістичних рішень. У цьому підрозділі буде проведено аналіз основних бездротових технологій, що застосовуються у складських середовищах – Wi-Fi, 5G, LoRaWAN, ZigBee, Bluetooth Mesh – з точки зору їхньої придатності для управління мобільними платформами та інтеграції в єдину систему IoT.

Мережева інфраструктура виконує низку ключових функцій:

- керування в реальному часі – WMS надсилає команди мобільним платформам (напрямок руху, швидкість, завдання);
- збір телеметрії – кожен робот передає дані про місцезнаходження, заряд акумулятора, стан сенсорів;
- підтримка навігації – системи SLAM і Lidar генерують потік даних, який необхідно оперативно передавати на центральний сервер або в хмару;
- відеопередача – роботи з камерами використовують відеопотік для аналізу середовища;
- координація між роботами – обмін даними між платформами для уникнення зіткнень або узгоджених дій.

Для реалізації цих задач мережа повинна забезпечувати наступні характеристики:

- низьку затримку – яка повинна бути не більше 10 – 20 мс для критичних команд;
- високу пропускну здатність;
- високу надійність і мінімальні втрати пакетів;
- плавний роумінг між точками доступу без розриву зв'язку;
- підтримку тисяч одночасних підключень.

Далі розглянемо основні технології, які здатні задовільнити ці вимоги.

Wi-Fi 6 та його розширена версія Wi-Fi 6E стали найпоширенішим рішенням для внутрішніх логістичних приміщень. Порівняно з попередніми поколіннями, вони забезпечують [13]:

- пропускну здатність до 9,6 Гбіт/с;

- зменшену затримку до 2-5 мс;
- підтримку до 1024 одночасних клієнтів на одній точці доступу;
- роботу у діапазонах 2.4 ГГц, 5 ГГц та 6 ГГц;
- нову технологію OFDMA (Orthogonal Frequency Division Multiple Access), яка ділить канал на підканали й дозволяє одночасну передачу даних кільком пристроям;
- механізм BSS Coloring, що знижує перешкоди при великій щільності клієнтів.

Wi-Fi 6 має наступні переваги в сфері логістики:

- висока пропускна здатність – дає змогу передавати дані з відеокамер, Lidar-сенсорів та систем SLAM без втрат;
- зниження затримки – важливо для точного позиціонування роботів;
- підтримка великої кількості роботів – склади можуть мати сотні AMR, і Wi-Fi 6 справляється з такою щільністю;
- зворотна сумісність із попередніми стандартами – дозволяє поступово оновлювати інфраструктуру;
- енергоефективність завдяки механізму TWT, який дозволяє зменшити споживання енергії пристроями, що працюють циклічно.

Але Wi-Fi 6 має і наступні недоліки, а саме основною його проблемою лишається роумінг – процес перемикання між точками доступу. Робот, який рухається 1 – 2 м/с, може перетинати зону покриття кілька разів на хвилину. При кожному переході можливе переривання зв'язку (100 – 500 мс), що здатне призвести до втрат пакета або затримки команди.

Отже, Wi-Fi 6/6E – основна технологія сучасних роботизованих складів. Вона забезпечує достатню швидкість і стабільність, але потребує грамотного планування інфраструктури, особливо для великих парків AMR.

Перейдемо до наступного виду мережі, а саме 5G, мережа яка стала справжнім проривом у промислових застосуваннях. Його ключова перевага – наднизька затримка та гарантована надійність передачі даних, що критично для систем керування у реальному часі.

5G використовує наступні три основні режими [14]:

- eMBB (Enhanced Mobile Broadband) – для високошвидкісної передачі даних;
- mMTC (Massive Machine Type Communications) – для великої кількості IoT-пристроїв;
- uRLLC (Ultra-Reliable Low-Latency Communications) – для критичних систем, таких як робототехніка.

Серед технологічних переваг даної мережі можна виділити наступне:

- наднизька затримка (< 1 мс) – дозволяє виконувати команди в реальному часі;
- безшовний роумінг – перемикання між базовими станціями відбувається миттєво, без втрат пакетів;
- Network Slicing – можливість «нарізки» мережі: окремі віртуальні сегменти для різних систем (роботи, відеоспостереження, адміністрація);
- підтримка до 10^6 пристроїв/км² – ідеально для великих складів;
- покриття великих площ із мінімальною кількістю антен.

Нині активно у промисловості впроваджуються приватні 5G-мережі – локальні мережі, які розгортаються без участі публічних операторів. Вони дають змогу:

- гарантувати повну конфіденційність даних;
- налаштувати власні параметри QoS (Quality of Service);
- інтегрувати мережу з існуючими системами автоматизації (PLC, WMS, SCADA);
- використовувати SIM-аутентифікацію для безпеки роботів.

Але і недоліки в 5G-мережі також присутні, а саме:

- висока вартість розгортання – обладнання, ліцензії, налаштування;
- необхідність спеціалізованих інженерів;
- підтримка пристроїв – не всі AMR-платформи мають модулі 5G.

Отже, 5G – технологія майбутнього складів, яка повністю вирішує проблеми Wi-Fi щодо затримки та роумінгу. Очікується, що протягом

найближчих 5 – 7 років приватні 5G-мережі стануть стандартом для великих автоматизованих логістичних центрів.

Розберемо ще дві мережі, які можна було б застосувати у логістичних системах. Одна з них LoRaWAN – протокол для бездротового зв'язку з великою дальністю та низьким енергоспоживанням. Даний протокол працює на неліцензованих діапазонах (868 МГц) і забезпечує [15]:

- радіус дії до 15 км (у відкритій місцевості);
- низьку швидкість передачі (від 0.3 до 50 кбіт/с);
- мінімальне енергоспоживання, що дозволяє датчикам працювати роками на одній батареї.

Серед переваг даного протоколу хотілося б виділити, те що він:

- чудово підходить для стаціонарних сенсорів;
- просте розгортання і невисока вартість;
- можливість охоплення великої площі одним шлюзом.

Також в нього є низка недоліків, а саме, надто мала пропускна здатність, велика затримка та одностороння передача.

В кінці можна дійти висновку що LoRaWAN непридатна для управління AMR, але її можна використовувати для допоміжних задач, серед яких можна виділити моніторинг температур й вологості, контроль стану дверей, воріт, холодильних камер, а також відстеження палет або контейнерів із низькою частотою оновлення даних.

Наступна мережа ZigBee/Z-Wave, яка відома передусім у «розумних будинках», однак застосовується і в логістиці. Їхню ключовою особливістю є mesh-топологія: кожен пристрій може ретранслювати сигнал іншого, утворюючи мережу без централізованої точки доступу [16].

Серед переваг можна виділити наступне:

- низьке енергоспоживання;
- автоматичне самовідновлення мережі при виході вузлів;
- низька вартість модулів і простота інтеграції.

Серед їх недоліків можна виділити: невеликий радіус дії, низька

швидкість передачі, не підходять для потокових критичних даних, обмежена кількість «стрибків» сигналу через вузли.

Їх застосовують для взаємодії між сусідніми роботами, для зв'язку між локальними сенсорами у конкретній зоні, для контролю освітлення, температури або безпеки.

Нижче наведено порівняльну характеристику всіх вищезгаданих мереж (табл. 1.1).

Таблиця 1.1 – порівняльні характеристики мережевих технологій

Технологія	Пропускна здатність	Затримка	Дальність	Енергоспоживання	Придатність для AMR
Wi-Fi 6/6E	до 9.6 Гбіт/с	2-5 мс	50-100 м	середнє	+
5G	до 10 Гбіт/с	< 1 мс	500-1000 м	середнє	+
LoRaWAN	0.3-50 кбіт/с	> 1000 мс	до 15 км	дуже низьке	-
ZigBee/Z-Wave	до 250 кбіт/с	10-100 мс	до 30 м	низьке	+ -

Жодна окрема технологія не може повністю задовольнити всі вимоги. Тому сучасні логістичні комплекси використовують гібридну мереживу модель, яка поєднує кілька рівнів зв'язку:

- рівень мобільних платформ (AMR) – використовується Wi-Fi 6 або 5G для основного зв'язку та управління. Забезпечує високу пропускну здатність і мінімальну затримку;

- рівень стаціонарних сенсорів – LoraWAN або ZigBee для збору повільних даних;

- локальна взаємодія роботів – ZigBee для короткодіючих каналів «робот

- робот»;

– хмарна інтеграція – через шлюзи Wi-Fi/5G дані передаються до WMS, ERP або хмарної аналітики.

Така архітектура дозволяє забезпечити стійкість, гнучкість та масштабованість, дозволяючи підтримувати одночасно тисячі пристроїв із різними вимогами до зв'язку.

Отже, можна зробити загальний висновок, що мережева інфраструктура є фундаментом роботизованої логістики. Вона визначає, наскільки точно і швидко система реагує на зміни, як координуються десятки або сотні AMR, і наскільки стабільною є робота складу в цілому.

Проведений порівняльний аналіз показав:

– Wi-Fi 6/6E – нинішній стандарт для більшості складів завдяки оптимальному співвідношенню швидкості, вартості та сумісності;

– 5G – перспективний напрямок для великих підприємств, що вимагають найвищої надійності та мінімальної затримки;

– LoRaWAN, ZigBee – допоміжні технології для сенсорних систем і локальних задач.

У результаті формується інтегрована багаторівнева мережева архітектура, у якій кожен рівень виконує свою роль у забезпеченні стабільності, швидкодії та безпеки роботи всіх складових системи.

1.4 Аналіз існуючих програмних рішень

Основним елементом управління у сучасній складській логістиці є ПЗ, яке працює від приймання товарів до їх відвантаження. Без чіткої координації між системами різних рівнів навіть найсучасніше обладнання та роботи не забезпечать ефективності. Із-за цього архітектура ПЗ складського комплексу зазвичай будується за ієрархічним принципом, де кожен рівень має власні функції, але водночас інтегрується з іншими для створення єдиної екосистеми управління.

Основними рівнями ПЗ є:

- WMS – верхній рівень системи управління складом;
- WES (Warehouse Execution System) – система виконання складських операцій, яка є середнім рівнем[17];
- FMS/RMC (Fleet Management System / Robot Management Console) – система управління парком мобільних роботів, яка є нижнім рівнем.

Саме ці три рівні і формують «мозок» сучасного складу, тим самим забезпечуючи як стратегічне планування, так і безпосереднє виконання завдань мобільними платформами.

WMS, яка є верхнім рівнем програмної ієрархії. Її можна розглядати як головний інтелектуальний центр складу, який відповідає за:

- управління запасами;
- облік приймання, розміщення, комплектації, відвантаження;
- планування робочих змін і ресурсів;
- формування звітів та аналітики;
- інтеграцію з ERP-системами підприємства.

WMS не керує роботами безпосередньо. Дана система працює на рівні логістичних задач – наприклад, створює завдання: «зібрати замовлення, яке складається з товарів А, Б, В». Далі ці завдання передаються вже на нижчі рівні – WES та FMS, які вже розподіляють їх між людьми, машинами або роботами.

Нижче буде наведено існуючі приклади систем WMS:

- SAP EWM – одне із найпотужніших корпоративних рішень, інтегроване в екосистему SAP ERP. Підтримує автоматизовані склади, штрих-кодування, RFID, 3D-візуалізацію складу;
- Oracle WMS cloud – хмарна система з широкими аналітичними можливостями, орієнтована на великі логістичні оператори;
- Manhattan Associates WMS – рішення преміум-класу з глибокою аналітикою та оптимізацією потоків;
- Qguar WMS – популярна в Європі система середнього рівня з гнучкою

інтеграцією з автоматизованими підсистемами;

– Infor WMS, HighJump, Fishbowl – інші поширені продукти з різним ступенем автоматизації.

Система WMS – це стратегічний рівень управління складом, який оперує бізнес-процесами, але не здійснює безпосереднє керування обладнанням чи роботами. Її ефективність залежить від здатності взаємодіяти з нижчими рівнями системи – WES і FMS.

Розберемо нижчий рівень WES, який є проміжним рівнем між стратегічним управлінням WMS та технічним управлінням FMS. Головною функцією системи є оперативна диспетчеризація процесів у реальному часі. Якщо WMS визначає що потрібно зробити, то WES вирішує як саме це зробити найефективніше.

Розглянемо основні функції даної системи:

- розподіл завдань між персоналом, конвеєрами та роботами;
- оптимізація черговості виконання замовлень;
- контроль завантаження кожної зони складу;
- балансування потоків товарів;
- інтеграція з сенсорами, сканерами, автоматизованими конвеєрними лініями;
- моніторинг виконання у реальному часі.

Таким чином, WES виступає операційним диспетчером, який здійснює управління всіма ресурсами складу.

Особливостями взаємодії з іншими рівнями можна виділити наступне:

- вхідна інформація: отримує завдання від WMS;
- вихідна інформація: формує конкретні інструкції для систем нижчого рівня;
- зворотній зв'язок: повідомляє WMS про статус виконання завдань, швидкість обробки, відхилення від плану.

Отже, WES можна назвати «операційною надбудовою», що забезпечує гнучке управління ресурсами складу. Без цього рівня робота автоматизованого

складу здійснювалася б фрагментарно, оскільки WMS не здійснює врахування поточного стану роботів, а FMS не має інформації про бізнес-пріоритети.

Далі розберемо найнижчий рівень ієрархії, а саме FMS, який безпосередньо взаємодіє з мобільними платформами. FMS здійснює фізичну реалізацію завдання, яке було визначено WES.

На нього полягають такі функції:

- призначення завдань – система обирає найближчого вільного робота, якого далі залучає для виконання місії;
- планування маршрутів – побудова оптимальної траєкторії з урахуванням поточного трафіку;
- управління трафіком – уникнення зіткнень, пріоритезація руху, динамічне коригування шляху;
- моніторинг стану роботів – батарея, сенсори, діагностика, технічне обслуговування;
- автоматична зарядка – направлення роботів до станцій заряджання;
- передача телеметрії – постійне оновлення координат і статусів до WES або WMS.

Розглянемо які проблеми має система, а саме проблема сумісності, а точніше відсутність універсальних стандартів інтеграції FMS. Більшість виробників створюють власні пропрієтарні системи, які можуть працювати лише з їхніми роботами. Як приклад:

- KION Group Fleet Controller – для роботів KION;
- Geek+ RMS – для AMR-платформ Geek+;
- Locus Robotics Console, MiR Fleet, GreyOrange Butler Manager, Omron Fleet Operations Workspace — подібні системи з обмеженою сумісністю.

Це створює проблему інтеграції: якщо склад використовує обладнання різних виробників, доводиться розгортати кілька незалежних FMS або створювати додатковий рівень інтеграції через API.

У відповідь на проблему сумісності почали з'являтися відкриті стандарти і платформи інтеграції:

- VDA 5050 – європейський стандарт обміну даними між різними AGV/AMR і системою управління;
- Mass Robotics Interoperability Standard – ініціатива для спільної роботи AMR різних брендів;
- Open Robot Communication Interface (ORiN) – універсальний протокол, що використовується в Японії;
- ROS 2 Industrial – відкрите середовище, яке дозволяє інтегрувати роботів різних виробників у спільну систему.

Таким чином, майбутнє логістики рухається у напрямку універсальної FMS-платформи, здатної керувати змішаним парком роботів через стандартизовані протоколи зв'язку.

Отже, FMS/RMC – це операційний центр управління роботами, який забезпечує безпечну, ефективну та скоординовану роботу мобільних платформ. Її розвиток безпосередньо визначає рівень автономності складу.

1.5 Постановка задачі на розробку програмного забезпечення

1.5.1 Формулювання задачі

Метою даної роботи є створення ПЗ для автономної мобільної платформи, орієнтованої на виконання логістичних операцій у складському середовищі, із використанням мережевих технологій та сучасних алгоритмів навігації. Система повинна забезпечувати взаємодію між серверною частиною та мобільним роботом через MQTT-брокер [18], що дозволяє організувати легку, масштабовану та надійну комунікацію у режимі реального часу.

Задача полягає у розробленні двох взаємопов'язаних компонентів:

- сервер управління мобільною платформою, який формує логістичні місії, аналізує телеметрію робота, контролює виконання навігаційних завдань, коригує траєкторію руху та забезпечує управління кількома роботами у майбутньому масштабуванні;

– контролер автономної мобільної платформи, який отримує команди від серверу, виконує навігаційні алгоритми, формує поведінкові реакції на перешкоди, збирає дані сенсорів, керує приводами та повертає телеметрію на сервер.

Загальна постановка задачі полягає у створенні системи, здатної:

- виконувати рух до цільових точок;
- уникати перешкоди у реальному часі;
- дотримуватися меж робочої зони;
- працювати у ручному та автономному режимах;
- обмінюватися даними із затримкою не більше 100 мс;
- точно повторювати маршрути, сформовані серверною частиною.

Створений комплекс має моделювати роботу реальної мобільної платформи у складі автоматизованої логістичної системи.

1.5.2 Функціональні вимоги

Функціональні вимоги визначають, що повинна вміти система та відображають кінцеву поведінку, очікувану від ПЗ.

Основні функціональні до розроблюваного ПЗ вимоги:

- підтримка ручного керування (режим LISTEN) – робот повинен реагувати на прямі команди оператора: лінійну та кутову швидкість;
- виконання автоматичних місій (режим AUTO) – платформа повинна досягати послідовності точок, підтримуючи задані параметри руху та коригуючи траєкторію;
- уникнення перешкод (режим AVOID) – реакція має базуватися на даних фронтального сонару з адаптивною поведінкою;
- контроль меж робочого поля (режим BOUNDARY) – у разі виходу за припустиму зону робот повинен автоматично змінити рух;

- збирання і передавання телеметрії – контролер повинен надсилати координати, орієнтацію, швидкість, стан сенсорів, активний режим роботи, підтвердження виконання місії;
- сумісність із симуляційним середовищем Webots[19] – модель робота повинна повністю повторювати логіку реального контролера;
- підтримка журанлювання – необхідно записувати дані для подальшого аналізу траєкторії та тестування алгоритмів.

1.5.3 Технічні вимоги

Технічні вимоги визначають параметри продуктивності із якими система повинна впевнено працювати.

Технічні вимоги та обмеження виглядають наступним чином:

- максимальна затримка MQTT-комунікації – до 100 мс, яка необхідна для реального часу та плавності руху;
- підтримка QoS 1 для комунікації – це гарантує доставку команд і телеметрії;
- масштабованість до кількох роботів, серверу необхідно працювати з довільною кількістю платформ без зміни архітектури;
- контролер повинен працювати, як у Webots, так і на реальному обладнанні;
- модулі серверу і контролера повинні бути незалежними;
- у випадку втрати зв'язку робот повинен переходити автоматично в безпечний режим.

1.5.4 Експлуатаційні вимоги

Експлуатаційні вимоги визначають поведінку ПЗ у реальних умовах роботи:

- безпечність руху – алгоритм повинен робити зупинку при небезпечній дистанції, об'їжджати перешкоду та запобігати зіткненням;

- автоматичне відновлення після збоїв, а саме у разі втрати комунікації рух має зупинятися до відновлення зв'язку;
- робот не повинен залишати визначених меж;
- система повинна працювати годинами без деградації;
- дані телеметрії мають бути доступні для аналізу.

1.5.5 Очікувані результати

У ході виконання очікується отримати, наступне:

- повноцінну клієнт-серверну систему для керування AMR;
- автономний навігаційний модуль із поведінковими режимами;
- модуль виконання місій із корекцією траєкторії;
- стійку систему MQTT-комунікації у режимі реального часу;
- симуляційну модель у Webots, що відповідає реальному роботу;
- систему запису та аналізу журналів руху;
- прототип програмного забезпечення, який можна переносити на фізичну платформу.

1.5.6 Наукова новизна роботи

Наукова новизна роботи полягає у розробленні інтегрованої клієнт-серверної системи управління мобільною платформою на основі MQTT та поведінкових алгоритмів, що у комплексі забезпечують функціонування автономного робота в умовах складської логістики.

Основні елементи наукової новизни:

- створено нову архітектурну модель розподіленої системи керування мобільною платформою;
- вперше запропоновано поєднання поведінкових режимів (AUTO, LISTEN, AVOID, BOUNDARY) у єдину адаптивну навігаційну систему;
- реалізовано серверний модуль оброблення місій із можливістю корекції траєкторії на основі аналізу телеметрії;

- удосконалено підхід до реактивної поведінки робота з використанням фільтрації сенсорних даних;
- застосовано метод аналізу телеметрії для оцінки та оптимізації параметрів управління;
- продемонстровано можливість перенесення ПЗ зі симуляції Webots на реальне апаратне забезпечення.

1.6 Висновок до розділу

Під час опрацювання другого розділу було проведено комплексний теоретичний аналіз сучасних тенденцій у сфері автоматизації, цифровізації та роботизації складських процесів, а також розглянуто архітектуру системи управління і мережеві технології, які забезпечують функціонування мобільних платформ. Отримані результати дослідження дають змогу сформуванню цілісного розуміння структури «розумного складу» в межах концепції Логістика 4.0, що базується на інтеграції кіберфізичних систем, Інтернету речей (IoT), Big Data, штучного інтелекту та автономних мобільних роботів.

В підрозділі 1.1 було розглянуто перехід від традиційної логістики до Логістики 4.0, що означає зміщення фокусу з ручних операцій на цифрові системи управління, які забезпечують прозорість, точність та можливість прогнозування. Цифровізація створює основу в той час, як автоматизація і роботизація реалізують фізичне виконання процесів. Їхнє поєднання формує замкнуту екосистему, у якій дані, обладнання та алгоритми взаємодіють у реальному часі.

В наступному підрозділі було розглянуто мобільні платформи, які є центральним елементом сучасного автоматизованого складу. Які ділять на два типи AGV та AMR. AGV залишаються надійними для стабільних процесів, але AMR стали стандартом гнучкої логістики, забезпечуючи самостійне планування маршрутів, динамічне ухилення від перешкод і взаємодію з WMS.

Далі було розглянуто мережеві технології, для ефективного керування парком AMR потрібна високошвидкісна, стабільна та безшовна комунікаційна інфраструктура, яка забезпечує обмін даними в реальному часі. В ході проведення порівняльного аналізу та розбору мереж. Зроблено висновок, що сучасні склади застосовують гібридну модель зв'язку, у якій Wi-Fi 6 або 5G використовуються для роботів, а LoRaWAN і ZigBee – для сенсорів. Таке поєднання дозволяє забезпечити оптимальний баланс швидкості, надійності та енергоефективності.

В кінці було здійснено аналіз існуючих програмних рішень, де було розглянуто, програмну інфраструктуру сучасного складу, яка побудована за трирівневою ієрархією, що визначає розподіл функцій між системами WMS, WES, FMS. Основною проблемою останнього є відсутність універсальних стандартів сумісності, адже більшість виробників пропонують власні пропрієтарні рішення, що обмежує інтеграцію змішаного парку роботів різних брендів.

У сукупності ці елементи формують інтелектуальну логістичну систему, що відповідає принципам Індустрії 4.0 – саморегульовану, гнучку й масштабовану систему, здатну працювати практично без участі людини.

Саме на основі отриманих теоретичних висновків формуються вимоги до розробки власного ПЗ для керування мобільною платформою, яке буде розглядатися в наступних розділах. Це ПЗ має забезпечити інтеграцію між рівнями WMS/WES/FMS, підтримувати роботу з різними мережевими технологіями та реалізовувати інтелектуальне управління автономним рухом у межах складського простору.

Наведено постановку задачі, визначено вимоги до програмного забезпечення, сформовано концепцію та структуру системи, а також викладено наукову новизну роботи. Встановлені функціональні, технічні та експлуатаційні параметри стали основою для проєктування та реалізації ПЗ, описаних у наступному розділі.

2 РОЗРОБЛЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Процес розробки ПЗ для керування мобільною платформою вимагав необхідної комбінації з раніше представленої теоретичної інформації, з минулих розділів, для практичної реалізації інструментів автоматичного керування, обміну даними та обробки інформації з датчиків.

Процес створення ПЗ складався з декількох етапів, а саме:

- вибір мов програмування та технологій;
- проектування архітектури системи;
- розроблення комунікаційного протоколу та структури повідомлень;
- реалізації контролера мобільної платформи;
- створення серверної частини для оброблення логістичних місій;
- інтеграцію та налагодження компонентів;
- тестування в симуляційному середовищі та аналіз отриманих результатів.

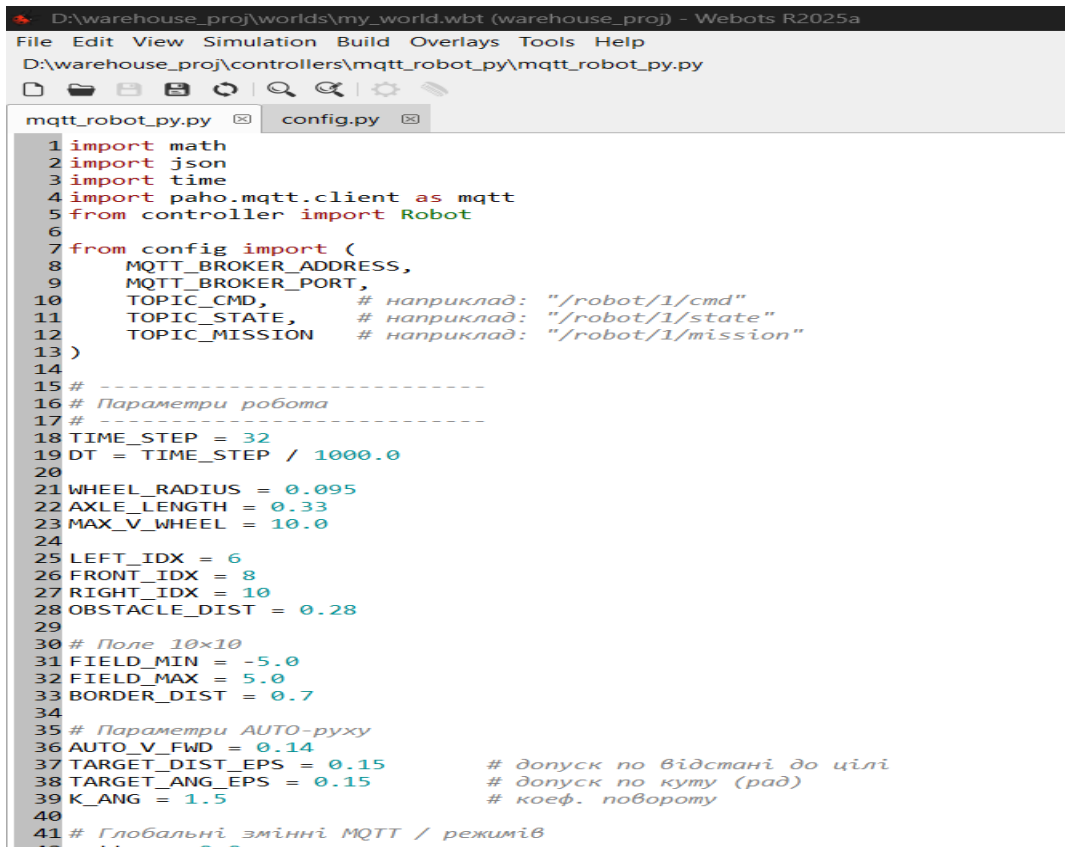
Кожен із даних етапів відіграє не аби яке важливе значення для того, щоб забезпечити коректну роботу мобільної платформи в автономному та дистанційному режимах.

2.1 Вибір мов програмування та програмних інструментів

Для створення нашої системи було обрано дві популярні мови програмування – Python [20] та C++ [21], оскільки вони оптимально доповнюють одна одну та відповідають вимогам систем автоматизованої логістики.

Python було обрано для розроблення контролера мобільної платформи, адже він підтримується симулятором Webots, як стандартна мова контролера, має велику кількість бібліотек для роботи з MQTT, забезпечує високу

швидкість розробки та простоту налагодження, дозволяє ефективно працювати з даними датчиків та теметрією. Фрагмент контролеру створеного на мові програмування Python наведено на рис. 2.1.



```

D:\warehouse_proj\worlds\my_world.wbt (warehouse_proj) - Webots R2025a
File Edit View Simulation Build Overlays Tools Help
D:\warehouse_proj\controllers\mqtt_robot_py\mqtt_robot_py.py

mqtt_robot_py.py  config.py
1 import math
2 import json
3 import time
4 import paho.mqtt.client as mqtt
5 from controller import Robot
6
7 from config import (
8     MQTT_BROKER_ADDRESS,
9     MQTT_BROKER_PORT,
10    TOPIC_CMD,          # наприклад: "/robot/1/cmd"
11    TOPIC_STATE,       # наприклад: "/robot/1/state"
12    TOPIC_MISSION      # наприклад: "/robot/1/mission"
13 )
14
15 # -----
16 # Параметри робота
17 # -----
18 TIME_STEP = 32
19 DT = TIME_STEP / 1000.0
20
21 WHEEL_RADIUS = 0.095
22 AXLE_LENGTH = 0.33
23 MAX_V_WHEEL = 10.0
24
25 LEFT_IDX = 6
26 FRONT_IDX = 8
27 RIGHT_IDX = 10
28 OBSTACLE_DIST = 0.28
29
30 # Поле 10x10
31 FIELD_MIN = -5.0
32 FIELD_MAX = 5.0
33 BORDER_DIST = 0.7
34
35 # Параметри AUTO-руху
36 AUTO_V_FWD = 0.14
37 TARGET_DIST_EPS = 0.15          # допуск по відстані до цілі
38 TARGET_ANG_EPS = 0.15          # допуск по куту (рад)
39 K_ANG = 1.5                     # коеф. повороту
40
41 # Глобальні змінні MQTT / режимів
42

```

Рисунок 2.1 – Контролер на мові Python

C++ було обрано для створення серверної логіки, яка обробляє логічні місії, з наступних причин:

- висока продуктивність при обчисленні навігаційних параметрів;
- стабільність роботи у режимі тривалих сеансів;
- широкі можливості для реалізації алгоритмів управління;
- сумісність із MQTT-клієнтами промислового стандарту.

Крім того для тесту розробленого нашого ПЗ було обрано Webots, який чудово підходить для створення симуляції робота та моделювання його поведінки у складських умовах, нижче буде наведено фрагмент серверу створеного на C++ рис 2.2., та середовище симуляції рис 2.2.

```

43     while (a > PI) a -= 2 * PI;
44     while (a < -PI) a += 2 * PI;
45     return a;
46 }
47
48 bool goToPoint(MQTTClientWrapper& mqtt, double gx, double gy, const std::string& l
49 const double Kv = 0.35;
50 const double Kw = 1;
51 const double max_v = 0.3;
52 const double max_w = 1.0;
53 const double tol = 0.25;
54
55 while (true) {
56     Pose p;
57     {
58         std::lock_guard<std::mutex> lk(poseMutex);
59         p = pose;
60     }
61
62     double dx = gx - p.x;
63     double dy = gy - p.y;
64     double dist = sqrt(dx * dx + dy * dy);
65
66     if (dist < tol) {
67         mqtt.publish(config::TOPIC_CMD, "{\"linear\":0,\"angular\":0}");
68         return true;
69     }
70 }

```

Рисунок 2.2 – фрагмент серверу створеного на С++

```

322 v_cmd = 0.0
323 w_cmd = 0.0
324
325 # -----
326 # Avoid obstacle (має працювати)
327 # -----
328 if front_f < OBSTACLE_DIST:
329     current_mode = "AVOID"
330     v_cmd = 0.0
331     w_cmd = 0.55
332
333 # -----
334 # Boundary protection
335 # -----
336 if (x < FIELD_MIN + BORDER_DIST or
337     x > FIELD_MAX - BORDER_DIST or
338     y < FIELD_MIN + BORDER_DIST or
339     y > FIELD_MAX - BORDER_DIST):
340
341     current_mode = "BOUNDARY"
342     v_cmd = -0.05
343     w_cmd = 0.4
344
345 # =====
346 # Wheel speeds
347 # =====
348 v1, vr = compute_wheel_speeds(v_cmd, w_cmd)
349 v1 = max(-MAX_V_WHEEL, min(MAX_V_WHEEL, v1))
350 vr = max(-MAX_V_WHEEL, min(MAX_V_WHEEL, vr))
351
352 left_motor.setVelocity(v1)
353 right_motor.setVelocity(vr)
354
355 # =====
356 # -----
357
358 state = {
359     "x": x,
360     "y": y,
361     "theta": yaw,
362     "mode": current_mode,
363 }

```

Рисунок 2.3 – середовище Webots

Отже, в цьому підрозділі було розглянуто обрані нами мови програмування та середовище симуляції для тестування роботи нашого ПЗ. Далі ми перейдемо до проектування архітектури програмної системи.

2.2 Проектування архітектури програмної системи

Ідея проектування архітектури ПЗ для керування мобільною платформою полягала на принципах модульності, масштабованості та незалежності компонентів. На основі здійсненого аналізу вимог та особливостей складських логістичних процесів було обрано клієнт-серверну архітектуру з асинхронним обміном повідомленнями, яка найбільш ефективно підтримує роботу автономних мобільних роботів у реальному симуляційному середовищі.

Архітектура ПЗ складається з трьох головних підсистем:

- контролер мобільної платформи;
- MQTT-брокер (комунікаційний вузол);
- сервер логістичних місій;

Кожен із цих компонентів виконує окремий набір функцій, котрі підвищують надійність і робить систему стійкою до помилок та зривів зв'язку.

Розберемо кожен компонент по порядку, контролер, який реалізовано на мові Python та інтегровано в симуляційне середовище Webots. Він несе відповідальність за наступне:

- отримання даних від датчиків (GPS, IMU, ультразвукові сенсори);
- оброблення навігаційних параметрів;
- виконання команд, отриманих від сервера місій;
- локальне прийняття рішень у режимі уникнення перешкод;
- передавання телеметрії на MQTT-брокер.

Контролер в системі є автономною одиницею, яка може функціонувати незалежно від серверної частини у разі тимчасового розриву зв'язку. Це відповідає перед вимогами реальних роботизованих систем, де безпека пересування не може залежати від мережевих затримок та відмов.

Наступний компонент це MQTT-брокер, який здійснює обмін інформацією між компонентами системи за допомогою однойменного протоколу, який широко використовується у промислових IoT-системах

завдяки низькому споживанню трафіку, мінімальним затримкам, підтримці великої кількості одночасних клієнтів та гарантії доставки повідомлень через механізм QoS.

MQTT є центральною точкою синхронізації, через яку усі компоненти обмінюються інформацією. Даний підхід дозволяє уникнути прив'язки одного модуля до іншого, використовувати будь-які платформи та мови програмування та легко розширювати систему додаванням нових роботі.

Для забезпечення логічного розділення потоків інформації було визначено перелік MQTT-топиків, а саме:

- /robot/1/cmd – команди управління рухом;
- /robot/1/state – передавання телеметрії від робота;
- /robot/1/mission – задачі, що надходять від серверної частини.

Наступним є сервер управління логістичними завданнями, який виконує функції з оброблення та валідації даних місії, керує чергою логістичних операцій, здійснює навігаційний супровід робота під час виконання місії, контролює досягнення ключових точок маршруту та здійснює формування команд руху на основі поточного стану платформи.

На відміну від контролера, сервер не виконує безпосереднє управління апаратними компонентами, що дозволяє легко замінювати або модернізувати робототехнічну платформу без зміни логістичної логіки.

Реалізація серверу на мові C++, забезпечує високу продуктивність, стабільність у багатопотокових обчисленнях та можливість запуску на промислових контролерах або edge-комп'ютерах.

Процес взаємодії компонентів виглядає наступним чином:

- сервер отримує завдання та додає його до черги;
- сервер обчислює першу точку та формує команду;
- команда передається через MQTT-топик /robot/1/cmd;
- контролер приймає команду та коригує швидкість і напрямок;
- робот передає телеметрію у топик /robot/1/state;

- сервер аналізує координати та визначає, чи досягнуто цілі;
- у разі успіху сервер переходить до наступної точки маршруту;
- після завершення місії сервер формує повідомлення про успішне виконання.

Дана схема дозволяє відокремити навігаційний блок від логістичного. Це є важливим принципом сучасних AMR-систем.

Отже, можна виділити наступні ключові переваги запропонованої архітектури:

- модульність – компоненти можуть оновлюватися незалежно, що дозволяє спростити модернізацію;
- масштабованість – система може працювати із будь якою кількістю роботів;
- універсальність – можлива інтеграція, як із симуляційним середовищем, так і з реальним апаратним забезпеченням;
- надійність – у разі втрати зв'язку робот зберігає здатність до безпечного пересування;
- низькі затримки – MQTT забезпечує реальній час взаємодії між компонентами.

Таким чином, можна дійти висновку, що обрана структура відповідає сучасним тенденціям автоматизації складської логістики, зокрема принципам Логістики 4.0 та кіберфізичних систем. Вона дозволяє відокремити високорівневу логіку від низькорівневого управління, забезпечити адаптивність алгоритмів навігації, використовувати цифровий двійник для тестування, а також перейти на реальну платформу без змін у програмній архітектурі. Все попередньо перераховане робить систему придатною для впровадження, як у навчальних, так і у промислових умовах.

2.3 Розроблення комунікаційного протоколу та структури інформаційних повідомлень

На основі проектної архітектури було сформовано вимоги до комунікаційного протоколу між мобільною платформою та сервером логістичних місій. Основними вимогами було забезпечити мінімальну затримку передачі команд, гарантувати доставку критично важливих повідомлень, задовільнити можливість масштабування кількості підключених роботів, а також незалежність компонентів та відсутність жорсткої прив'язки між ними. Враховуючи перераховані вимоги, було обрано протокол MQTT (Message Queuing Telemetry Transport), який належить до легковагових протоколів передачі повідомлень та є стандартом де-факто в IoT-системах.

Вимоги до розробленого протоколу:

- однозначність інтерпретації команд – кожен пакет даних повинен мати чітку структуру та ключі;
- розділення каналів (топіків) – для запобігання змішуванню команд та станів робота;
- можливість розширення – формат повідомлень має передбачати додавання нових параметрів без зміни існуючих модулів;
- стійкість до втрати зв'язку – робот повинен зберігати працездатність навіть у разі тимчасового розриву каналу.

Структуру MQTT-топіків наведено нижче у табл 2.1.

Таблиця 2.1 – структура MQTT-топіків

Топік	Напрямок	Призначення
/robot/1/cmd	сервер ->робот	Передавання команд руху
/robot/1/state	робот -> сервер	Телеметрія робота
/robot/1/mission	сервер -> робот	Передавання місій та маршрутів

Дане розділення дозволяє керувати кількома роботами одночасно, вести незалежні потоки даних та легко масштабувати систему.

Для уніфікації використано формат JSON, що дозволяє забезпечити простоту аналізу, універсальність, читабельність та підтримку у більшості мов програмування.

На рівні протоколу було сформовано кілька типів повідомлень, типу команд, телеметрії, місій, системних повідомлень.

Нижче наведено приклад розширеного телеметричного повідомлення:

```
{  
  "timestamp": 1731573021,  
  "x": -1.24,  
  "y": 2.57,  
  "yaw": -0.45,  
  "mode": "AUTO",  
  "sensors": {  
    "front": 0.78,  
    "left": 0.94,  
    "right": 1.05  
  },  
  "v_cmd": 0.12,  
  "w_cmd": 0.31  
}
```

Це дозволяє серверу не лише відстежувати положення робота, а й контролювати його стан, навантаження, режим роботи та поведінку, щодо перешкод.

2.4 Реалізація контролера мобільної платформи

Ключовим елементом системи є контролер мобільної платформи, оскільки саме він забезпечує інтерпретацію команд, отриманих від серверної

частини, оброблення сенсорної інформації та формування керуючих сигналів для руху робота. Його реалізація виконана мовою Python, що було обрано через гнучкість, високу швидкість розробки та тісну інтеграцію з симуляційним середовищем Webots. Структура контролера побудована модульно, що дає змогу легко адаптувати та масштабувати систему під різні типи роботизованих систем платформ, а також переносити програмний код із симуляції на реальне апаратне забезпечення з мінімальними змінами.

На початку роботи контролер завантажує основні кінематичні параметри мобільної платформи, зокрема радіус коліс, базову відстань між ними на занчення максимальних швидкостей. Такі параметри винесено в окрему секцію, що забезпечує можливість швидкої калібрації. Всі режими роботи робота (ручний, автономний, режим уникання перешкод та режим захисту меж поля) також визначені через глобальні змінні, що формують поточний стан машини. Такий підхід дозволяє контролеру оперативно реагувати на зміни умов середовища та коригувати поведінку.

У процесі роботи контролер використовує низку допоміжних математичних функцій. Серед них – функції згладжування даних сенсорів та нормалізації кутів, що забезпечують стабільність роботи алгоритмів навігації. Зокрема, для уникнення помилок, пов'язаних зі стрибками кута між значенням 180 та -180 градусів, використовується спеціальна функція нормалізації:

```
def angle_wrap(a):  
    while a > math.pi: a -= 2 * math.pi  
    while a < -math.pi: a += 2 * math.pi  
    return a
```

Завдяки цьому контролер коректно визначає напрямок руху та кутову похибку.

Однією з найважливіших функцій контролера є взаємодія з MQTT-брокером. Контролер підписується на теми команд керування та місій і прослуховує вхідні повідомлення у форматі JSON. У разі отримання команди

типу manual робот переходить у режим прямого підпорядкування оператору.

Наприклад:

```
if data.get("type") == "manual":
    current_mode = "LISTEN"
    mqtt_v = data["linear"]
    mqtt_w = data["angular"]
```

Це дозволяє оператору дистанційно керувати роботом у режимі реального часу, що є необхідним як для тестування, так і для аварійних ситуацій.

Якщо робот отримує завдання типу auto, контролер завантажує послідовність точок місії та переходить у режим автономного руху.

Наступним етапом є ініціалізація сенсорів та виконавчих механізмів. Через API Webots контролер підключається до GPS, IMU та сонарів. Двигуни коліс переводяться у режим безконечного обертання, що дозволяє задавати їм довільні швидкості. Такий спосіб керування є типовим для реальних диференціальних роботів.

Алгоритм навігації в автономному режимі побудований за принципом послідовного руху до цільових точок місії. Після зчитування телеметрії контролер порівнює поточні координати робота з координатами цілі, визначає необхідний кут руху та обчислює кутову похибку:

```
heading_err = angle_wrap(target_angle - yaw)
v_cmd = AUTO_V_FWD
w_cmd = K_ANG * heading_err
```

Таким чином робот рухається за принципом пропорційного регулятора (P-контролера), який забезпечує плавне та стабільне коригування траєкторії.

Важливим елементом контролера є реалізація режимів безпеки. Якщо фронтальний сонар реєструє небезпечно малу дистанцію, контролер переходить у режим AVOID, зупиняє поступальний рух та виконує маневр ухилення:

```
if front_f < OBSTACLE_DIST:
    current_mode = "AVOID"
```

```
v_cmd = 0
```

```
w_cmd = 0.5
```

Якщо робот наближається до меж робочого поля, активується режим BOUNDARY, який забезпечує повернення машини у безпечну зону. Такі режими моделюють базову поведінкову рефлексію, характерну для сучасних автономних роботів промислового класу.

Весь цикл роботи контролера виконується у реальному часі, а всі сформовані команди приводу обчислюються з урахування фізичних можливостей платформи. Після визначення лінійної та кутової швидкостей контролер здійснює їх перетворення у швидкості коліс і надсилає сигнал приводам.

Окрім керування, контролер забезпечує постійне публікування телеметрії у відповідну MQTT-тему. Структура повідомлення містить координати робота, його орієнтацію, активний режим та сенсорні значення. Наприклад:

```
state = {"x": x, "y": y, "mode": current_mode}
client.publish(TOPIC_STATE, json.dumps(state))
```

Також реалізовано локальне журналювання у CSV-файли, що дозволяє здійснювати поглиблений аналіз роботи алгоритмів, оцінювати точність руху та проводити калібрування навігаційних параметрів.

Узагальнюючи, контролер мобільної платформи поєднує в собі сенсорну обробку, високорівневу навігацію, поведінкову логіку, механізми безпеки та комунікаційні протоколи. Його архітектура є модульною, гнучкою та масштабованою, що дозволяє використовувати його як у симуляції Webots, так і на реальних автономних роботах у середовищах складської логістики.

2.5 Реалізація серверної частини системи управління логістичними місіями

Серверна частина системи виконує функцію центрального координатора, який забезпечує управління логістичними місіями та взаємодією з автономною

мобільною платформою. На відміну від контролера робота, який зосереджується на локальному керуванні та роботі з датчиками, сервер відповідає за глобальну логіку, прийняття рішень на рівні задач, обробку телеметрії, контроль виконання маршруту та забезпечення узгодженості всієї системи. Для реалізації серверного компонента було обрано мову програмування C++, яка забезпечує високу продуктивність, детермінованість та низькі затримки при обробці даних. Такі властивості є критично важливими при створенні систем керування мобільними роботами у реальному часі.

Після запуску сервер встановлює підключення до MQTT-брокера та підписується на канал телеметрії мобільної платформи. Це дозволяє отримувати безперервний потік даних про положення робота, його орієнтацію, стан системи та інші параметри, що необхідні для аналізу поточної ситуації. Взаємодія з брокером інкапсульована у спеціалізований клас, що спрощує логіку основного коду і дозволяє обробляти повідомлення у форматі JSON. Отримані дані сервер декодує, після чого оновлює внутрішню модель стану робота.

Приклад обробки вхідного повідомлення:

```
json state = json::parse(msg);  
double x = state["x"];  
double y = state["y"];  
double yaw = state["theta"];
```

Сервер зберігає у своїй структурі чергу логістичних точок, які формують поточну місію. Кожна точка містить координати, яких робот повинен досягнути. Черга дозволяє визначити порядок виконання задач, забезпечуючи їх послідовну обробку. Такий підхід надає змогу формувати складні маршрути, що складаються з багатьох точок, характерних для складської логістики.

Після оновлення телеметрії сервер перевіряє, чи є активна місія, і чи залишилися точки, які необхідно виконати. Якщо такі є, то сервер виконує навігаційні розрахунки: визначає зміщення між поточним положенням робота та ціллю, обчислює відстань та необхідний напрямок руху. Далі формується

кутова похибка, яка є основою для подальшого визначення лінійної та кутової швидкостей. Такий підхід заснований на класичних методах керування мобільними роботами, що використовують різницю між бажаним та фактичним напрямком руху як сигнал управління.

Фрагмент коду, що демонструє такі обчислення:

```
double dx = target.x - pos.x;
double dy = target.y - pos.y;
double target_angle = atan2(dy, dx);
double heading_err = angle_wrap(target_angle - pos.yaw);
```

На основі цих величин сервер формує команду для робота. Якщо кутова похибка значна, робот спочатку коригує орієнтацію, виконуючи поворот; якщо орієнтація достатньо точна — сервер задає рух уперед. Такий двофазний підхід підвищує точність досягнення цільових точок, що підтверджується практичним використанням аналогічних алгоритмів у промислових AMR-системах.

Приклад формування керуючої команди:

```
json cmd;
cmd["linear"] = v;
cmd["angular"] = w;
mqtt.publish(config::TOPIC_CMD, cmd.dump());
```

Сервер відстежує момент досягнення точки. Якщо відстань між роботом і ціллю стає меншою за встановлений поріг, вважається, що точка виконана, після чого вона вилучається із черги, а сервер переходить до наступної. Такий механізм дозволяє гнучко підтримувати маршрути будь-якої довжини. У випадку, якщо черга вичерпана, сервер завершує місію та надсилає відповідний статус.

Окреме місце у роботі сервера займають механізми контролю помилок. Якщо протягом певного інтервалу часу сервер не отримує телеметрію, або якщо робот не здійснює руху попри наявність активних команд, сервер переводить систему у стан помилки та повідомляє про це оператора. Подібні ситуації можуть свідчити про втрату зв'язку, перепони, які робот не може подолати, або

про апаратні збої. Впроваджена система контролю забезпечує стабільність та безпеку функціонування всієї робототехнічної платформи.

Узагальнюючи, серверна частина виконує роль координатора, що приймає рішення на основі даних телеметрії та логістичних вимог. Вона забезпечує логіку руху, коригування траєкторії, аналіз помилок і роботу з MQTT-комунікацією. Обрана архітектура дозволила досягти надійності, зрозумілості та простоти масштабування, а також забезпечила можливість перенесення системи з симуляційного середовища у реальні логістичні умови з мінімальними змінами.

2.6 Інтеграція контролера та серверної підсистеми. Імітація роботи мобільної платформи

Після розроблення окремих програмних компонентів — контролера мобільної платформи та серверної частини управління місіями — наступним етапом стала інтеграція цих підсистем та проведення їх комплексного тестування у симуляційному середовищі Webots. Метою інтеграції було перевірити коректність взаємодії всіх модулів, оцінити стійкість комунікації через MQTT-брокер, дослідити поведінку роботи в різних експлуатаційних умовах та підтвердити працездатність розроблених алгоритмів автономної навігації.

Основою інтеграції є асинхронна комунікаційна модель, у межах якої сервер та контролер працюють незалежно один від одного, обмінюючись даними через брокер MQTT. Такий підхід відповідає сучасним принципам розподілених кіберфізичних систем і дозволяє масштабувати систему для роботи з багатьма мобільними роботами одночасно. Контролер, працюючи на боці робота, постійно надсилає телеметричні дані, які містять точні координати, орієнтацію та сенсорні значення. Сервер у свою чергу аналізує отриману інформацію, оновлює внутрішню модель стану платформи та формує відповідні керуючі команди.

Імітація роботи починається із запуску контролера у Webots. Робот проходить стадію ініціалізації, перевіряє доступність сенсорів і встановлює зв'язок із MQTT-брокером. Одночасно запускається серверна частина, яка очікує на перші телеметричні повідомлення та формування місії. Після отримання координат робот офіційно вважається «активним», і сервер отримує можливість надсилати команди управління або завдання на автономне переміщення.

Типовий цикл взаємодії контролера та сервера у симуляції виглядає таким чином:

- оператор надсилає місію у вигляді набору точок;
- сервер приймає місію, додає її до внутрішньої черги та обчислює першу ціль;
- контролер отримує команди керування, інтерпретує їх і задає лінійну та кутову швидкості;
- робот рухається у напрямку цілі, використовуючи вбудовану поведінкову логіку для корекції руху;
- телеметрія повертається на сервер, який на її основі оновлює параметри команди;
- при досягненні точки сервер переводить робота до наступної цілі.

Повідомлення у процесі взаємодії мають компактний JSON-формат. Наприклад, телеметрія, що надсилається з частотою до 10 Гц, може мати вигляд:

```
{"x": 1.24, "y": -0.85, "theta": 0.45, "mode": "AUTO"}
```

Команди руху, що формуються сервером, передаються у такому ж структурованому форматі:

```
{"linear": 0.12, "angular": 0.32}
```

Під час інтеграційних випробувань значну увагу приділяли аналізу затримок у передаванні повідомлень. Вимірювання показали, що система стабільно працює з мережею затримок у межах 4–12 мс, що повністю відповідає вимогам реального автономного транспорту у складських

приміщеннях. Цей результат підтвердив доцільність вибору MQTT як основного протоколу комунікації.

Важливою частиною інтеграційного тестування було моделювання аварійних сценаріїв та поведінки робота у разі перешкод. Симуляція показала, що контролер коректно реагує на небезпечне наближення до об'єктів: він переходить у режим уникнення, припиняє поступальний рух та виконує маневр обертання, доки дистанція не стане безпечною. Так само успішно функціонував механізм захисту меж поля, який запобігає виходу платформи за встановлену зону.

Після кожного тестового запуску відбувався запис траєкторій у CSV-журнали для подальшого аналізу. Це дозволило побудувати точні траєкторії руху, оцінити відхилення від оптимального маршруту, визначити похибки навігаційних алгоритмів та вдосконалити параметри контролера. Наприклад, типовий запис журналу має вигляд: 1731542042,1.24,-0.85,0.44,AUTO,0.12,0.31.

Дані з журналів дали змогу підтвердити, що середня похибка позиціонування під час тестів становила 0.12–0.18 м, що є цілком прийнятним показником для роботів класу AMR (Autonomous Mobile Robots), особливо у складських умовах із невисокою динамікою руху.

Проведені випробування також продемонстрували надійність алгоритмів перемикання між режимами роботи. Навіть за умов переривання зв'язку або наявності перешкод система стабільно поверталася до основного сценарію виконання місії після усунення проблеми. Це свідчить про високу стійкість архітектури та правильне проектування механізмів безпеки у контролері.

Імітація роботи мобільної платформи в цілому показала, що розроблена система здатна розв'язувати логістичні задачі, динамічно адаптуватися до поточного стану середовища, підтримувати стабільну комунікацію та виконувати складні маршрути. Таким чином, інтеграційні випробування підтвердили практичну працездатність створеного програмного забезпечення та його готовність до подальшого перенесення на реальну мобільну платформу. Результати, отримані в процесі тестування, демонструють, що архітектурні

рішення, використані в роботі, є обґрунтованими та ефективними.

2.7 Тестування та аналіз результатів роботи системи

Метою тестування є перевірка працездатності розробленого ПЗ для керування мобільною платформою, а також оцінювання відповідності його функціональних, технічних та експлуатаційних характеристик вимогам, сформульованим у розділі постановка задачі. Тестування дозволяє підтвердити коректність реалізованих алгоритмів, надійність мережевої взаємодії та можливість практичного застосування в умовах складської логістики.

Випробування створеної програмної системи проводилося в середовищі Webots, яке забезпечує моделювання фізичних характеристик мобільної платформи використовувався протокол MQTT, що дозволило перевірити роботу клієнт-серверної архітектури в режимі, який є близьким до реального часу.

2.7.1 Методологія тестування

Тестування здійснювалося поетапно з урахуванням основних режимів роботи мобільної платформи та типових сценаріїв її експлуатації. Загальний вигляд тестового середовища та розміщення мобільної платформи в симуляторі наведено на рисунку 2.4.

У процесі дослідження було виконано наступні заходи:

- функціональне тестування режимів керування;
- тестування стабільності MQTT-комунікацій;
- тестування алгоритмів автономної навігації;
- тестування поведінки системи у нештатних ситуаціях;
- стрес-тестування при інтенсивному обміні даними.

Кожен тестовий сценарій було потворено декілька разів, що дозволило оцінити відтворюваність результатів та стабільність роботи системи.

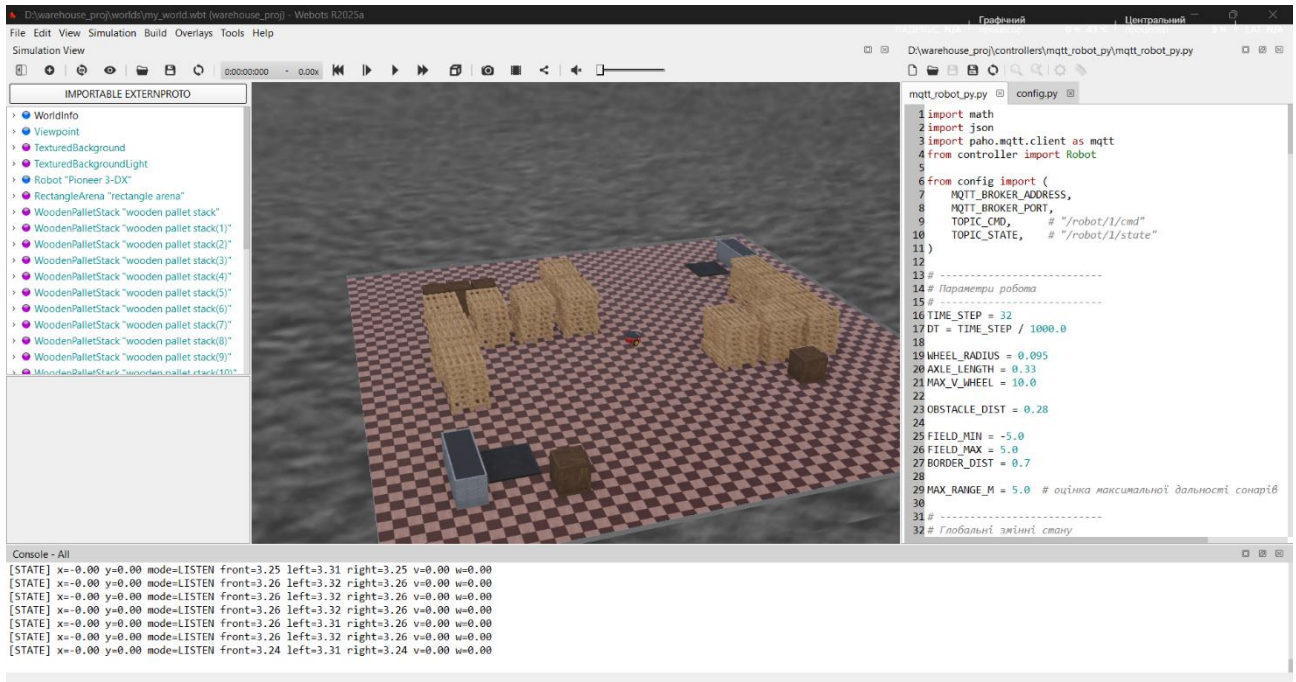


Рисунок 2.4 – Загальний вигляд тестового середовища у симуляторі Webots

2.7.2 Опис тестових сценаріїв

Під час експериментальних досліджень було реалізовано кілька основних тестових сценаріїв.

Перший тестовий сценарій спрямований на перевірку коректності роботи ручного керування мобільною платформою. У даному режимі керуючі команди формувалися на серверній стороні або за допомогою зовнішнього клієнта MQTT та передавалися безпосередньо контролеру платформи.

У процесі тестування здійснювалося поступове передавання команд мобільній платформі, а саме лінійна та кутова швидкість. В ході проведення аналізу результатів побачено, що платформа стабільно реагує на зміну параметрів керування, рухається у заданому напрямку та коректно виконує обертання навколо власної осі. Затримка між надсиланням команди та реакцією платформи є незначною і не впливає на керованість руху.

Даний сценарій дозволив підтвердити коректність реалізації базового циклу керування, обміну даними через MQTT та кінематичної моделі мобільної

платформи. Рух мобільної платформи в режимі ручного керування продемонстровано на рисунку 2.5.

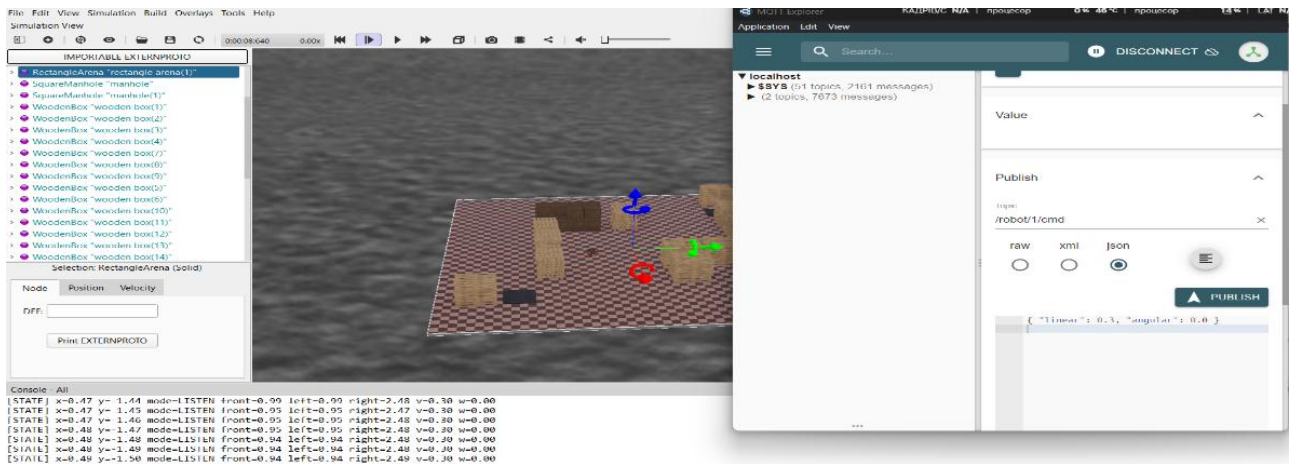


Рисунок 2.5 – Рух мобільної платформи в режимі ручного керування та обмін MQTT-повідомленнями

Наступний сценарій передбачає собою тестування автономного режиму роботи мобільної платформи. Платформі задається маршрут, який складається з кількох послідовних цільових точок. Дані координат точок передавалися серверній частині, яка формувала керуючі команди для контролера платформи.

Під час виконання сценарію платформа сама обчислює напрямок руху до потенційної точки, коригує швидкість та орієнтацію і помірно досягає кожен з них. Досягнення цільових точок супроводжується зменшенням швидкості та стабілізацією положення платформи.

У процесі тестування зафіксовано, що платформа рухається за плавною траєкторією без різких змін напрямку руху, а похибка позиціонування перебуває в межах допустимих значень. Даний сценарій підтвердив працездатність алгоритмів автономної навігації та коректність взаємодії між сервером і контролером платформи. Даний сценарій продемонстровано на рисунку 2.6.

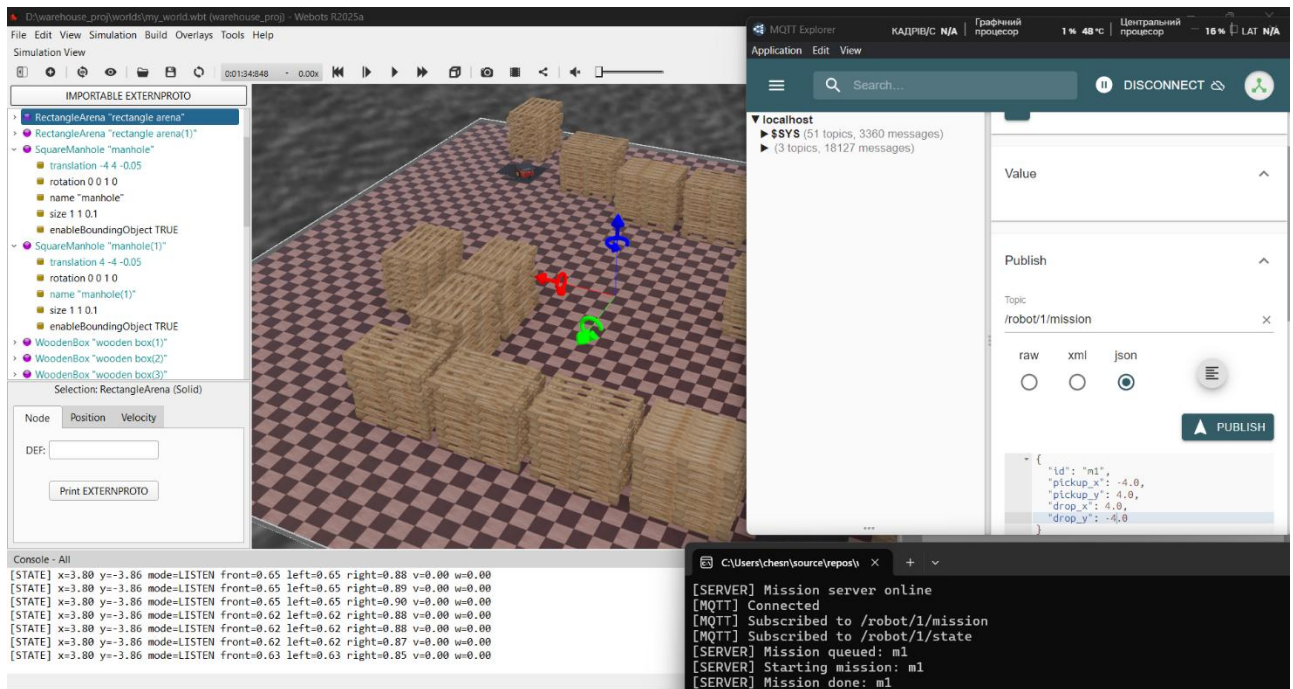


Рисунок 2.6 – Рух мобільної платформи за заданим маршрутом

Третій тестовий сценарій спрямований на перевірку алгоритмів уникнення перешкод. На траєкторії руху мобільної платформи розміщувалися штучні перешкоди, які фіксувалися сенсорними пристроями платформи, так званими сонарами.

Після виявлення перешкод система автоматично переходила у режим уникнення, зменшувала лінійну швидкість та формувала коригуючі команди для уникнення перешкоди. Після успішного маневру платформа поверталася до виконання основного маршруту.

Результати тестування показали, що алгоритм уникнення перешкод працює стабільно, а платформа не допускає зіткнень навіть при повторному наближенні до об'єктів. Це підтверджує ефективність реалізованих поведінкових механізмів та їх придатність для використання у складському середовищі. Перехід платформи в режим уникнення перешкод продемонстровано на рисунку 2.7.

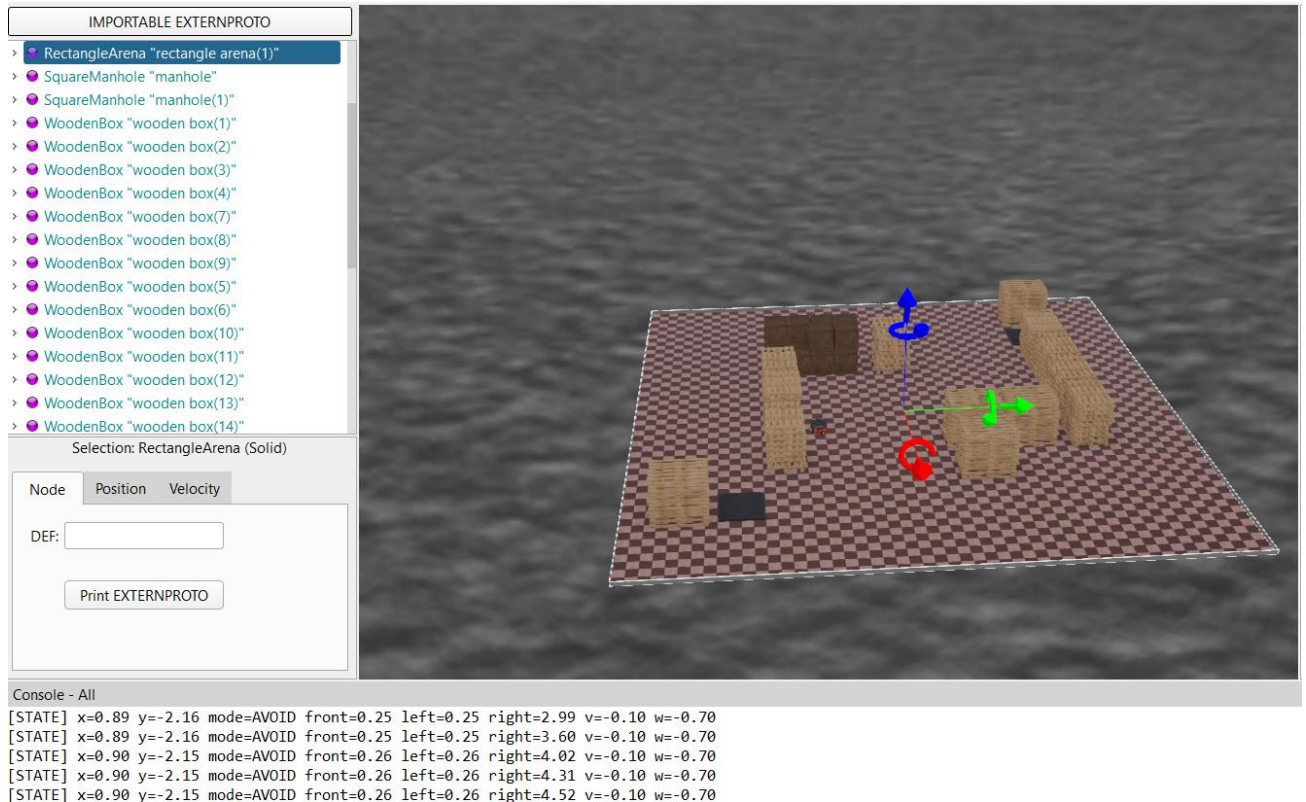


Рисунок 2.7 – режим уникнення перешкоди

Під час тестування четвертого сценарію, а саме контролю платформою меж робочої зони, було наближено її до заданих меж допустимого простору або надано команди руху у напрямку виходу за ці межі.

У разі виявлення порушення система автоматично змінювала режим роботи, зменшувала швидкість руху та виконувала маневр повернення у дозволу зону. Така поведінка забезпечує додатковий рівень безпеки та запобігає виходу платформи за межі робочого середовища. Режим контролю меж робочої зони продемонстровано на рисунку 2.8.

Сценарій 5 – Втрата мережового з'єднання.

Останній тестовий сценарій передбачає імітацію втрати мережового з'єднання між серверною частиною та контролером мобільної платформи. У процесі тестування припинявся обмін MQTT-повідомленнями, що дозволило оцінити поведінку в нештатній ситуації.

У результаті чого платформа припиняла виконання активних команд та переходила у безпечний стан, що виключає можливість неконтрольованого

руху. Після відновлення з'єднання система могла бути повторно переведена у робочий режим без необхідності перезапуску симуляції.

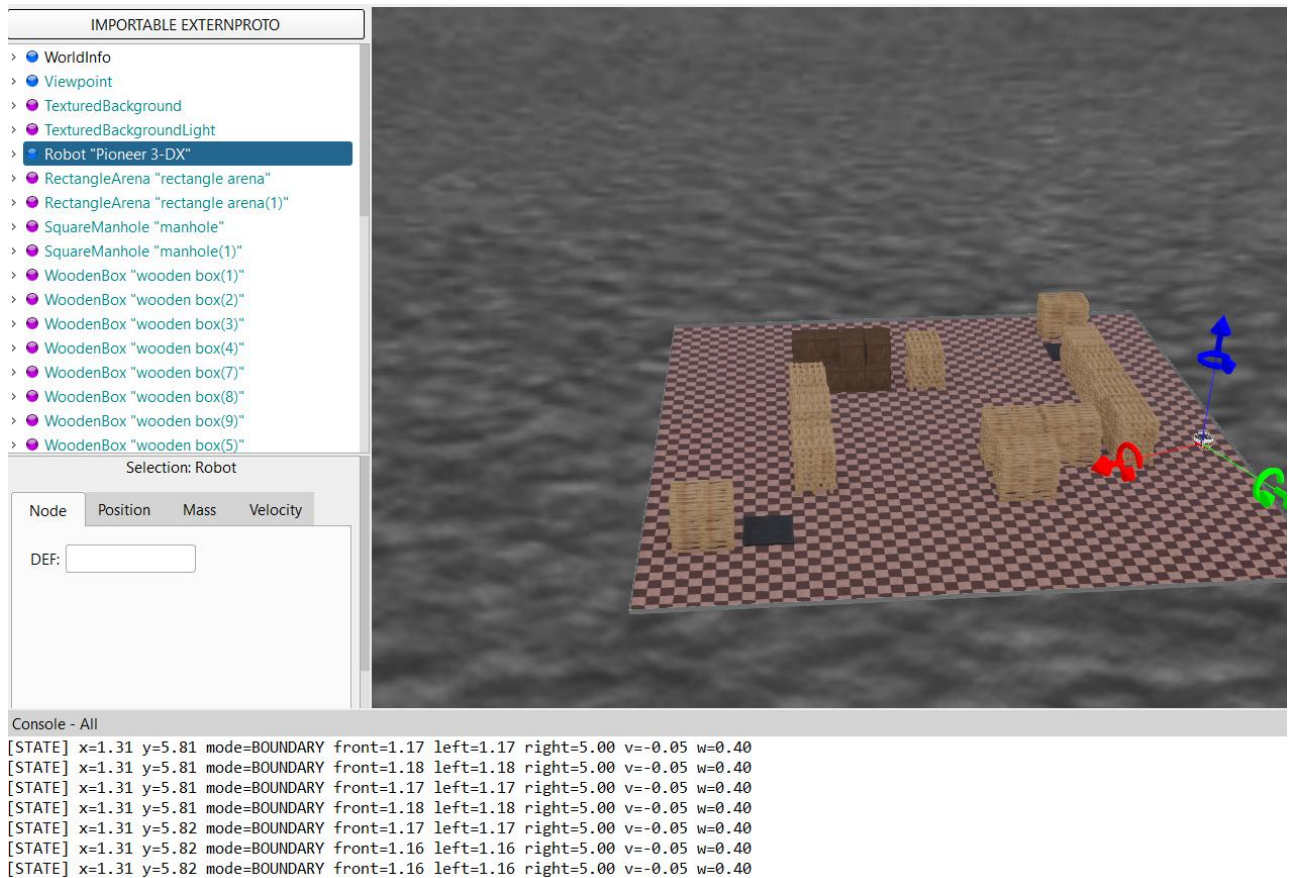


Рисунок 2.8 – режим контролю меж робочої зони

Проведені тестові сценарії підтвердили, що розроблене ПЗ коректно реалізує всі передбачені режими роботи мобільної плаформи. Система демонструє стабільну поведінку, здатність адаптуватися до змін умов середовища та безпечну роботу у нештатних ситуаціях, що є критично важливим для застосування в умовах складської логістики.

2.7.3 Аналіз результатів тестування

За результатами виконання тестових сценаріїв, описаних в минулому підрозділі, було проведено аналіз кількісних та якісних показників роботи розробленої програмної системи керування мобільною платформою. Найбільше

уваги приділялося оцінці стабільності мереживої взаємодії, точності навігації та швидкості системи в різних режимах роботи.

Під час тестування ручного та автономного режимів керування було зафіксовано затримку передавання MQTT-повідомлень, яка у середньому перебувала в межах від 0 до 4 мс. Отримані значення можна пояснити локальним характером розгортання системи та відсутністю мережових затримок, притаманних реальним розподіленням системам. У реальних умовах експлуатації в локальній або бездротовій мережі очікується збільшення затримки до значень порядку 4-12 мс, що є допустимими значеннями для задач керування мобільними платформами складської логістики. Характер зміни затримки передавання повідомлень наведено на рисунку 2.9.



Рисунок 2.9 – Графік затримки передавання MQTT-повідомлень під час тестування

У процесі виконання автономних місій було оцінено точність досягнення заданих цільових точок маршруту. Аналіз показав, що середня похибка позиціонування мобільної платформи перебувала в межах 0,12-0,18 м, що є допустимими значеннями для задач складської логістики. Похибка орієнтації платформи не перевищувала 3 – 5°, що забезпечувало стабільний та передбачений рух під час навігації.

Окремо було проаналізовано стабільність роботи системи під час виконання сценаріїв уникнення перешкод та контролю меж робочої зони. У всіх випадках система коректно переходила між режимами роботи без втрати керованості та не допускала зіткнень або виходу платформи за межі дозволеної зони.

Частота виконання основного циклу керування контролером мобільної платформи становила близько 20-30 Гц, що дозволяло своєчасно обробляти сенсорні дані та формувати керуючі сигнали. Це підтверджує відповідність системи вимогам до програмного забезпечення реального часу.

Узагальнені результати наведені в таблиці 2.2.

Таблиця 2.2 – Основні кількісні показники роботи програмної системи

Показник	Значення
Затримка MQTT-повідомлень	0 – 4 мс
Похибка позиціонування	0,12 – 0,18 м
Похибка орієнтації	3 – 5°
Частота циклу керування	20 – 30 Гц

2.8 Висновок до розділу

В третьому розділі було розроблено та реалізовано програмне забезпечення для керування мобільною платформою на базі мережевих технологій, призначене для використання в умовах складської логістики. У процесі роботи визначено архітектуру програмної системи, обґрунтовано вибір мов програмування та інструментальних засобів, також було реалізовано клієнт-серверну модель керування з використанням протоколу MQTT.

У розділі було детально описано реалізацію контролера мобільної платформи та серверної частини системи з наведенням невеликих фрагментів коду. Контролер забезпечує обробку сенсорних даних, формування керуючих

сигналів та реалізацію основних режимів роботи платформи, зокрема ручного керування, автономного руху, уникнення перешкод і контролю меж робочої зони. Серверна частина відповідає за обробку місій, формування керуючих команд та організацію мережевої взаємодії між компонентами системи.

Особливо було приділено увагу інтеграції програмних компонентів у симуляційному середовищі Webots, що дозволило відпрацювати логіку керування мобільною платформою в умовах, наближених до реальних. Реалізована система забезпечує стабільний обмін даними між сервером та контролером, а також коректну взаємодію з навколишнім середовищем.

У межах розділу було проведено тестування розробленого ПЗ з використанням типових експлуатаційних сценаріїв. Результати тестування підтвердили працездатність і надійність системи, а також високу швидкість мережевої взаємодії. Затримка передавання керуючих MQTT-повідомлень у симуляційному середовищі з локальним брокером перебувала в межах 0-4 мс, що дозволяє здійснювати керування мобільною платформою в режимі, близькому до реального часу.

Отримані результати свідчать про відповідність розробленого ПЗ вимогам, які було сформульовані в розділі 2. Розроблена система може бути використана як основа для подальшого перенесення на реальну мобільну платформу та впровадження в системах складської логістики з урахуванням мережевих умов.

3 ТЕХНІКО-ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ ВПРОВАДЖЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Обґрунтування доцільності автоматизації складських операцій

Складська логістика характерна тим, що має велику кількість повторюваних операцій, пов'язаних із транспортуванням вантажів між зонами приймання, зберігання та відвантаження. У традиційних складських значна частина таких операцій виконується вручну, що призводить до підвищених витрат на оплату праці, зниження продуктивності та зростання впливу людсьУ сучасних умовах розвитку логістичних систем спостерігається тенденція до зростання обсягів вантажообігу та ускладнення внутрішньоскладських процесів. Це призводить до необхідності впровадження автоматизованих рішень, здатних забезпечити високу швидкість обробки матеріальних потоків, зменшення кількості помилок та підвищення загального рівня безпеки.

Особливо актуальною є автоматизація внутрішньо складського транспорту, оскільки саме на цю частину логістичного процесу припадає значна частка часу та витрат. Використання мобільних роботизованих платформ дозволяє оптимізувати ці процеси та створити передумови для подальшої цифровізації складської інфраструктури.кого фактора.

Автоматизація транспортних операцій з використанням мобільних платформ дозволяє зменшити навантаження на персонал, підвищити швидкість виконання операцій та забезпечити стабільну якість роботи незалежно від зовнішніх факторів. Ключовим елементом таких систем є ПЗ керування, яке забезпечує координацію руху платформи, її взаємодію з навколишнім середовищем та інтеграцію з інформаційними системами складу.

Розроблене в межах роботи ПЗ базується на використанні мережевих технологій і дозволяє реалізувати гнучку та масштабовану систему керування мобільною платформою.

3.2 Вихідні дані для економічних розрахунків

Для оцінки економічної ефективності впровадження розробленої системи приймемо такі умовні, але реалістичні вихідні дані:

- середня заробітна плата складського працівника – $Z = 18000$ грн/міс;
- кількість працівників задіяних у транспортних операціях – $N = 2$;
- кількість робочих місяців на рік – $M = 12$;
- вартість мобільної платформи – $C_{\text{тп}} = 120000$ грн;
- вартість впровадження та налаштування ПЗ – $C_{\text{пз}} = 20000$ грн;
- середній строк експлуатації системи – 5 років.

Наведені вихідні дані є усередненими та відповідають типовим умовам функціонування середнього складського комплексу. Вони дозволяють отримати орієнтовну оцінку економічної ефективності впровадження автоматизованої системи без прив'язки до конкретного підприємства.

Такий підхід є доцільним на етапі проектування та обґрунтування доцільності впровадження, оскільки дозволяє порівнювати різні варіанти автоматизації та оцінювати їх вплив на економічні показники діяльності підприємства.

3.3 Розрахунок витрат на ручне виконання складських операцій

Річні витрати на оплату праці персоналу, задіяного у виконанні транспортних операцій, визначається за формулою:

$$C_{\text{ручн}} = Z \cdot N \cdot M, \quad (3.1)$$

Підставивши значення отримаємо:

$$C_{\text{ручн}} = 18000 \cdot 2 \cdot 12 = 432000 \text{ грн/рік}$$

Отримане значення характеризує щорічні витрати підприємства лише на виконання транспортних операцій вручну без урахування додаткових витрат, пов'язаних із помилками, простоем або перевтомою персоналу.

3.4 Розрахунок витрат на автоматизований варіант

Загальні витрати на впровадження автоматизованої системи визначаються як сума вартості апаратної платформи та витрат на розроблення і провадження ПЗ:

$$C_{\text{авт}} = C_{\text{тр}} + C_{\text{пз}}, \quad (3.2)$$

$$C_{\text{авт}} = 120000 + 20000 = 140000 \text{ грн.}$$

Експлуатаційні витрати автоматизованої системи значно нижчі та включають витрати на електроенергію, технічне обслуговування та підтримку ПЗ. Для спрощення розрахунків прийmemo їх на рівні 10% від початкових витрат на рік:

$$C_{\text{експл}} = 0,1 \cdot C_{\text{авт}} = 14000 \text{ грн/рік}, \quad (3.3)$$

3.5 Розрахунок економічного ефекту від впровадження

Річний економічний ефект від впровадження мобільної платформи визначається, різниця між витратами на ручне виконання та експлуатаційними витратами автоматизованої системи:

$$E_{\text{річн}} = C_{\text{ручн}} - C_{\text{експл}}, \quad (3.4)$$

$$E_{\text{річн}} = 43200 - 14000 = 418000 \text{ грн/рік.}$$

Отриманий річний економічний ефект свідчить про значну фінансову доцільність впровадження мобільної платформи. Важливо зазначити, що у розрахунках не враховано непрямі економічні вигоди, такі як зменшення кількості помилок при транспортуванні вантажів, зниження рівня виробничого травматизму та скорочення часу простою обладнання.

Крім того, автоматизація транспортних операцій позитивно впливає на організацію логістичних процесів у цілому, забезпечуючи більш прогнозований і керований характер роботи складу.

3.6 Розрахунок терміну окупності системи

Термін окупності визначається за формулою:

$$T_{\text{ок}} = \frac{C_{\text{авт}}}{E_{\text{річн}}}, \quad (3.5)$$

$$T_{\text{ок}} = \frac{140000}{418000} \approx 0,34 \text{ року.}$$

Невеликий термін окупності системи є важливим показником її інвестиційної привабливості. Це дозволяє підприємству швидко компенсувати початкові витрати та надалі отримувати стабільний економічний ефект від експлуатації автоматизованої системи.

У разі масштабування системи та використання кількох мобільних платформ економічний ефект може зростати пропорційно обсягам автоматизованих операцій, що додатково підвищує доцільність впровадження розробленого програмного забезпечення.

3.7 Додаткові переваги впровадження системи

Окрім безпосереднього економічного ефекту, впровадження мобільної платформи з розробленим ПЗ забезпечує низку додаткових переваг:

- зниження впливу людського фактора;
- підвищення безпеки виконання складських операцій;
- можливість масштабування системи;
- інтеграцію з інформаційними системами складу;
- підвищення загальної продуктивності логістичних процесів.

Використання мережевих технологій керування забезпечує гнучкість системи та дозволяє адаптувати її до змін умов експлуатації без суттєвих витрат.

3.8 Висновки до розділу

У даному розділі було виконано техніко економічне обґрунтування впровадження ПЗ для керування мобільною платформою у складській логістиці. Проведені розрахунки показали, що автоматизація транспортних операцій дозволяє суттєво зменшити витрати на оплату праці та підвищити ефективність роботи складу.

Розрахований економічний ефект і малий термін окупності підтверджують доцільність впровадження розробленої системи. Отримані результати свідчать про перспективність використання мобільних платформ з мережевим керуванням у сучасних складських логістичних системах.

4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ

4.1 Загальна характеристика умов праці та нормативно-правова база

Використання мобільних роботизованих платформ у складській логістиці суттєво змінює характер умов праці персоналу та висуває підвищені вимоги до забезпечення охорони праці та безпеки життєдіяльності. Автоматизація транспортних операцій дозволяє зменшити фізичне навантаження на працівників, знизити вплив людського фактору та підвищити загальну ефективність логістичних процесів. Водночас поява рухомих автоматизованих засобів у робочому просторі складу створює нові потенційні небезпеки, які потребують системного аналізу та відповідних заходів безпеки.

Експлуатація мобільних платформ і ПЗ керування повинна здійснюватися відповідно чинного законодавства України у сфері охорони праці. Основними нормативно-правовими документами є Закон України «Про охорону праці», правила безпечної експлуатації електроустановок споживачів, державні санітарні норми і правила, а також нормативи пожежної безпеки для виробничих і складських приміщень. Дотримання зазначеній вимог є обов'язковою умовою забезпечення безпечних умов праці та запобігання виникненню аварійних і надзвичайних ситуацій.

4.2 Аналіз небезпечних і шкідливих виробничих факторів

Під час експлуатації мобільної платформи у складському середовищі персонал може зазнавати впливу різних небезпечних і шкідливих виробничих факторів. Найбільш характерними є механічні, електричні, фізичні та психофізіологічні фактори.

Механічні фактори пов'язані з рухом мобільної платформи в межах складу. До них належать можливі зіткнення платформ з працівниками,

стелажами або іншими об'єктами, а також ризик затискання між рухомою платформою та нерухомими конструкціями. Небезпека зростає в умовах обмеженого простору, інтенсивного руху та недостатнього зонування складських приміщень.

Електричні фактори зумовлені наявністю джерел електроживлення, електродвигунів, акумуляторів та серверного обладнання. Ураження електричним струмом можливе у разі пошкодження ізоляції, порушення правил експлуатації або проведення технічного обслуговування без відключення живлення.

Фізичні фактори включають шум і вібрації, що виникають під час роботи електроприводів мобільної платформи, а також вплив параметрів мікроклімату та освітлення складських приміщень. Недостатнє освітлення або підвищений рівень шуму можуть негативно впливати на працездатність персоналу та підвищувати ризик помилок.

Психофізіологічні фактори пов'язані з підвищеною відповідальністю оператора за контроль роботи автоматизованої системи, необхідністю тривалої концентрації уваги та роботою з комп'ютерною технікою. Тривале навантаження може призводити до втоми, зниження уваги та погіршення самопочуття.

4.3 Організаційні та технічні заходи забезпечення безпеки

Для зниження рівня небезпеки та забезпечення безпечної експлуатації мобільної платформи необхідно реалізувати комплекс організаційних і технічних заходів охорони праці.

До організаційних заходів належать проведення первинного, повторного та позапланового інструктажу з охорони праці, допуск до роботи лише навченого та проінструктованого персоналу, а також чітке зонування складських приміщень із виділенням зон руху мобільних платформ. У робочих зонах повинні бути встановлені попереджувальні знаки та нанесена відповідна

розмітка.

Технічні заходи безпеки реалізуються як на апаратному рівні так і на програмному. До них належать обмеження максимальної швидкості руху мобільної платформи, використання сенсорів для виявлення перешкод, реалізація аварійного зупинення та автоматичний перехід системи в безпечний режим у разі втрати мережевого з'єднання або програмного збою. ПЗ відіграє ключову роль у забезпеченні безпеки, оскільки дозволяє йому контролювати поведінку платформи в режимі реального часу та запобігати виникненню небезпечних ситуацій.

4.4 Вимоги електробезпеки та пожежної безпеки

Електробезпека є одним із найважливіших аспектів охорони праці при експлуатації мобільної роботизованої платформи та серверного обладнання. Усі електричні компоненти повинні відповідати встановленим вимогам безпеки, мати надійну ізоляцію та захист від короткого замикання. Забороняється експлуатація обладнання з пошкодженими кабелями, роз'ємами або контактами, а також виконання ремонтних робіт під напругою.

Для забезпечення пожежної безпеки необхідно дотримуватися правил експлуатації електрообладнання, не допускати перевантаження електричних мереж та використовувати лише сертифіковані електротехнічні компоненти. Складські приміщення повинні бути обладнані первинними засобами пожежогасіння, а персонал має бути ознайомлений з правилами дій у разі виникнення пожежі або іншої надзвичайної ситуації.

4.5 Ергономічні та санітарно-гігієнічні вимоги

Оператор системи керування мобільною платформою працює з комп'ютерною технікою, що потребує дотримання ергономічних і санітарно-гігієнічних вимог. Робоче місце повинно бути організоване таким чином, щоб

забезпечити зручну позу оператора, правильне розташування монітора, клавіатури та інших засобів введення інформації.

Освітлення робочого місця має відповідати встановленим нормам і забезпечувати достатній рівень видимості без утворення відблисків. Рекомендується дотримуватися режиму праці та відпочинку, робити регулярні перерви під час тривалої роботи за комп'ютером для зниження навантаження на зір та опорно-руховий апарат.

4.6 Дії персоналу в аварійних ситуаціях та висновки до розділу

У разі виникнення аварійної ситуації персонал повинен діяти відповідно до затверджених інструкцій з охорони праці. Основними діями є негайна зупинка мобільної платформи за допомогою аварійної кнопки, відключення електроживлення, повідомлення відповідальних осіб та, у разі необхідності, застосування первинних засобів пожежогасіння і евакуація персоналу.

Проведений у розділі аналіз показав, що впровадження мобільної роботизованої платформи за умови дотримання організаційних, технічних і програмних заходів безпеки дозволяє забезпечити належний рівень охорони праці та безпеки життєдіяльності. Запропоновані заходи відповідають чинним нормативним вимогам і підтверджують можливість безпечної експлуатації розробленої системи в умовах складської логістики.

ВИСНОВКИ

У ході виконання роботи було вирішено актуальну наукову-практичну задачу розроблення ПЗ для керування мобільною платформою на базі мережових технологій у складській логістиці. Актуальність теми зумовлена зростанням обсягів складських операцій, необхідністю автоматизації внутрішньоскладського транспорту та підвищення вимог до ефективності й безпеки логістичних процесів.

У процесі роботи було проаналізовано сучасні підходи до автоматизації складської логістики та використання мобільних роботизованих платформ. Проведений аналіз показав доцільність застосування клієнт-серверної архітектури керування з використанням мережових технологій, що забезпечує гнучкість, масштабованість і можливість інтеграції з іншими інформаційними системами підприємства.

У межах роботи сформульовано вимоги до ПЗ керування мобільною платформою, визначено основні функціональні режими роботи та обґрунтовано вибір інструментальних засобів і мов програмування. Для реалізації системи використано мови програмування C++ та Python, протокол MQTT і симуляційне середовище Webots, що дозволило відпрацювати логіку керування в умовах, наближених до реальних.

Розроблено архітектуру програмної системи керування, яка включає серверну частину та контролер мобільної платформи. Серверна частина забезпечує формування місій, обробку телеметричних даних і керування рухом платформи, тоді як контролер реалізує виконання керуючих команд, обробку сенсорної інформації та взаємодію з навколишнім середовищем. Реалізовано режими ручного керування, автономного руху, уникнення перешкод і контролю меж робочої зони.

У симуляційному середовищі Webots проведено тестування розробленого програмного забезпечення з використанням типових експлуатаційних сценаріїв.

Результати тестування підтвердили працездатність і стабільність системи, коректність реалізації алгоритмів керування та здатність системи адаптуватися до змін умов середовища. Експериментально встановлено, що затримка передавання керуючих MQTT-повідомлень у локальному середовищі перебуває в межах 0 – 4 мс, що забезпечує керування мобільною платформою в режимі, близькому до реального часу.

Виконано техніко-економічне обґрунтування впровадження розробленої системи. Проведені розрахунки показали, що автоматизація транспортних операцій за допомогою мобільної платформи дозволяє суттєво зменшити експлуатаційні витрати та забезпечити короткий термін окупності системи. Це підтверджує економічну доцільність використання розробленого програмного забезпечення у складських логістичних системах.

Розглянуто питання охорони праці та безпеки життєдіяльності при експлуатації мобільної роботизованої платформи. Проаналізовано небезпечні та шкідливі виробничі фактори, а також запропоновано комплекс організаційних, технічних і програмних заходів, спрямованих на забезпечення безпечних умов праці персоналу. Запропоновані заходи відповідають чинним нормативним вимогам та забезпечують можливість безпечного практичного використання розробленої системи.

Отримані в роботі результати підтверджують, що поставлена мета досягнута, а сформульовані завдання – виконані. Розроблене програмне забезпечення може бути використане як основа для подальшого перенесення на реальну мобільну платформу, масштабування системи та впровадження в реальних умовах складської логістики.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. ДСТУ 3008:2015 Інформація та документація «Звіти у сфері науки і техніки». Структура та правила оформлювання. / В. Земцева; Ю. Поліщук, канд. фіз.-мат. наук; Р. Санченко, канд. техн. наук; Л. Шрамко; А. Ямчук (науковий керівник) ДП «УкрНДНЦ» від 22 червня 2015р. № 61 з 2017-07-01.
2. Методичні вказівки з підготовки та захисту кваліфікаційної роботи здобувачами другого (магістерського) рівня вищої освіти спеціальності 174 Автоматизація, комп'ютерно-інтегровані технології та робототехніка, освітньо-професійних програм: «Комп'ютерно-інтегровані технологічні процеси і виробництва», «Комп'ютеризовані та робототехнічні системи» /Упоряд. І. Ш. Невлюдов, Р. В. Артюх, В. В. Безкоровайний, Н. П. Демська, В. В. Євсєєв, О. І. Филипенко, О. М. Цимбал. Харків: ХНУРЕ, 2022. – 49 с.
3. Положення про кваліфікаційну роботу здобувача вищої освіти на другому (магістерському) рівні [Електронний ресурс]: Наказ ХНУРЕ від 06 травня 2021 р. No 142. – Режим доступу: https://nure.ua/wpcontent/uploads/Main_Docs_NURE/143-vid-06.04.2021-pro-vvedennja-v-dijurishennja-vchenoi-radi-universitetu.pdf.
4. Жалдак М. І., Касьянов В. М. Інформаційні технології управління виробничими процесами. – Київ: Ліра-К, 2018. – 260 с.
5. Літвінов С. М. Бази даних і знань: теорія та проектування інформаційних систем. – Київ: Видавництво КНТ, 2019. – 344 с.
6. Катсанюк С. М., Лисенко В. А. Моделювання та алгоритми цифрових систем керування. – Київ: КНУТД, 2019. – 284 с.
7. Seborg D., Edgar T., Mellichamp D. Process Dynamics and Control. 4th ed. – Wiley, 2016. – 492 p.
8. Ковалишин В. В. Методи аналізу та моделювання динамічних систем. – Львів: Видавництво ЛНТУ, 2018. – 264 с.

9. Куваєв О. М., Литвиненко І. В. Математичне моделювання технічних систем. – Київ: Видавництво КПІ, 2020. – 356 с.
10. Катасонов Н. А. Математичне моделювання складних систем: аналітичні методи. – Харків: ХНАМГ, 2020. – 336 с.
11. Зайченко Ю. П. Дослідження операцій: моделі та методи. – Київ: КНЕУ, 2020. – 312 с.
12. Кузнецов М. Ю. Інтелектуальні системи керування: алгоритми та програмні засоби. – Харків: ХНУРЕ, 2022. – 312 с.
13. MQTT Version 4.1.1. OASIS Standard. – 2014. – Режим доступу: <https://mqtt.org>.
14. Python Software Foundation. Python 3 Documentation. – 2024. – Режим доступу: <https://docs.python.org/>.
15. Огарков С. Л. Математичні моделі в системах автоматизації: методи побудови і дослідження. – Львів: ЛНУ, 2021. – 298 с.
16. Пархоменко О. Ф. Методи оброблення даних у системах автоматизації. – Львів: ЛНУ, 2020. – 276 с.
17. Сорокоумова Н. В., Сухоруков М. Г. Теорія автоматичного керування: Математичні моделі та методи аналізу. – Харків: ХНУРЕ, 2019. – 412 с.
18. Комплекс навчально-методичного забезпечення навчальної дисципліни «Організація керування умовами праці» підготовки освітнього рівня бакалавр усіх спеціальностей та усіх напрямів університету [Електронний ресурс] / ХНУРЕ; розроб.: Т. Є. Стиценко, Г. В. Пронюк, Н. М. Сердюк. – Харків, 2017. – 108 с.