

Я, Царьова Аліна Олександрівна, як здобувачка вищої освіти ХНУРЕ, розумію і підтримую політику закладу із академічної доброчесності. Я не надавала і не одержувала недозволену допомогу під час підготовки кваліфікаційної роботи. Я не використовувала штучний інтелект для підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

"12" червня 2025 р



Аліна ЦАРЬОВА

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Факультет _____ АКТ _____
Кафедра _____ КІТАР _____
Рівень вищої освіти _____ перший (бакалаврський) _____
Спеціальність 151 Автоматизація та комп'ютерно-інтегровані технології
(код і повна назва)
Тип програми _____ Освітньо-професійна _____
Освітня програма Автоматизація та комп'ютерно-інтегровані технології
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри КІТАР _____
(підпис)
« 28 » квітня 2025 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Здобувачеві _____ Царьовій Аліні Олександрівні _____
(прізвище, ім'я, по батькові)

1. Тема роботи Розроблення системи інтелектуального керування робототехнічною платформою для виконання виробничих завдань

затверджена наказом по університету від 19.05.2025 р. № 390 Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 23.06.2025 р.

3. Вихідні дані до роботи _____

Платформа розроблена для інтеграції контурів управління сервоприводами, генерації траєкторій, програм на рівні завдань, програм з графічним інтерфейсом і 3D-моделювання в однорідну програмну архітектуру. Використовується тільки одна апаратна платформа, одна операційна система (QNX Real-Time Platform) і одна мова програмування (C++).

4. Перелік питань, що потрібно опрацювати в роботі _____

4.1 Аналіз технічного завдання _____

4.2 Розробка структурної схеми інтелектуальної системи управління _____

4.3 Реалізація структури інтелектуальної системи управління _____

4.4 Розробка роботизованої платформи _____

4.5 Написання, компіляція, зв'язок та запуск програм керування роботом _____

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Графічний матеріал у вигляді презентації, файл формату *.ppt, 13 арк. Ф. 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	Аналіз технічного завдання	28.04-04.05.2025	виконано
2	Розробка структурної схеми інтелектуальної системи управління	05.05-10.05.2025	виконано
3	Реалізація структури інтелектуальної системи управління	18.05-25.05.2025	виконано
4	Розробка роботизованої платформи	26.05-08.06.2025	виконано
5	Написання, компіляція, зв'язок та запуск програм керування роботом	09.06-11.06.2025	виконано
6	Подання роботи на перевірку Інтернет-сервісом StrikePlagiarism	12.06-15.06.2025	виконано
7	Подання роботи на рецензію	16.06-17.06.2025	виконано
8	Подання роботи на підпис зав. кафедри	18.06-19.06.2025	виконано
9	Подання кваліфікаційної роботи в ЕК	23.06.2025	виконано

Дата видачі завдання 28 квітня 2025 р.

Здобувач _____ Аліна ЦАРЬОВА
(підпис) (власне ім'я, прізвище)

Керівник роботи _____ доц. Наталія ДЕМСЬКА
(підпис) (посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка: 78 с., 30 рис., 1 додатків, 58 джерел.

КОЛІСНА РОБОТИЗОВАНА ПЛАТФОРМА, СИСТЕМА УПРАВЛІННЯ НЕЧІТКОЮ ЛОГІКОЮ, ІНТЕЛЕКТУАЛЬНІ БЕЗДРОТОВІ ДАТЧИКИ, НАБІР НЕЧІТКИХ ПРАВИЛ

Мета роботи – розробка програмного засобу для взаємодії інтелектуальних датчиків, машин та інших роботизованих систем, з метою створення високовзаємопов’язаного та автоматизованого виробничого середовища.

Об’єкт розробки – комплексні програмні рішення, що використовують алгоритми машинного навчання для аналізу даних, прийняття рішень та виконання завдань, що зазвичай потребують людського інтелекту .

Предмет розробки – робототехнічна платформа із інтелектуальним керуванням.

У даній роботі розроблено інтелектуальну систему управління рухом мобільної роботизованої платформи на основі комплексного підходу, яка охоплює: методи навігації, способи попередньої обробки даних; сучасні методи та алгоритми інтелектуального управління та нечіткої логіки; сучасна елементна база; методи інтелектуальної обробки та оцінки даних з датчиків в умовах перешкод і неповноти інформації; методи та засоби автоматизованого проектування апаратних і програмних засобів.

Розроблено структуру інтелектуальної системи управління рухом для колісної мобільної роботизованої платформи. Наведено результати моделювання руху колісної роботизованої платформи з різними значеннями показань з датчиків у навколишньому середовищі на основі розроблених правил. Запропоновано виконувати програмну розробку системи керування

мобільною роботизованою платформою у віддаленому режимі.

ABSTRACT

Explanatory note: 79 p., 29 figures, 1 appendices, 58 sources.

WHEEL ROBOTIC PLATFORM, FUZZY LOGIC CONTROL SYSTEM, INTELLIGENT WIRELESS SENSORS, SET OF FUZZY RULES

The purpose of the work is development of a software solution to enable interaction among intelligent sensors, machines, and other robotic systems, with the goal of establishing a highly interconnected and automated manufacturing environment.

The object of development is intelligent control of the robotic platform.

The subject of development is integrated software solutions employing machine learning algorithms for data analysis, decision-making, and the execution of tasks that typically require human intelligence.

In this work, an intelligent system for controlling the movement of a mobile robotic platform is developed based on an integrated approach, which includes: navigation methods, data preprocessing methods; modern methods and algorithms of intelligent control and fuzzy logic; modern element base; methods of intelligent processing and evaluation of data from sensors in conditions of obstacles and incomplete information; methods and tools for automated design of hardware and software.

The structure of an intelligent motion control system for a wheeled mobile robotic platform has been developed. The results of modeling the motion of a wheeled robotic platform with different values of sensor readings in the environment based on the developed rules are presented. It is proposed to perform software development of a mobile robotic platform control system in remote mode.

ЗМІСТ

Перелік скорочень	8
Вступ	9
1 Аналіз предметної області та літератури	11
1.1 Аналіз предметної області	11
1.1.1 Підключена автоматизація	13
1.1.2 Автономна навігація	16
1.1.3 Співпраця людини і машини	18
1.1.4 Гнучкі виробничі системи	22
1.1.5 Сучасні складальні лінії Intelligent Robotic Systems	25
1.1.6 Контроль та перевірка якості	25
1.1.7 Прийняття рішень на основі даних	26
1.1.8 Енергоефективність в інтелектуальних робототехнічних системах	28
1.1.9 Робота інтелектуальних систем у різних середовищах	28
1.2 Аналіз технічного завдання	29
1.2.1 Аналіз літератури	29
1.2.2 Аналіз технічного завдання	31
2 Розробка структурної схеми інтелектуальної системи управління	35
2.1 Розробка структури ІСУР	35
2.2 Алгоритми нечіткої логіки та засоби керування	38
3 Реалізація структури інтелектуальної системи управління	43
4 Розробка роботизованої платформи	53
4.1 Визначення архітектури	53
4.2 Визначення інструментів розробки	54
4.2.1 Робоча станція	54
4.2.2 Платформа реального часу QNX	55
4.2.3 Об'єктно-орієнтоване програмування на C++	55
4.2.4 Анімація об'єктів	56

4.2.5 Використання системи QMotor	56
4.3 Компоненти роботизованої платформи	56
4.4 Основні класи	60
4.5 Класи, пов'язані з маніпуляторами	61
4.6 Класи кінцевих ефекторів	63
4.7 Модель паралелізму	64
4.8 Менеджер об'єктів	65
4.9 Реалізація керування за допомогою QMotor	66
4.10 Математична бібліотека	67
4.11 Графічний інтерфейс користувача	69
4.12 Написання, компіляція, зв'язок та запуск програм керування роботом	70
4.13 Охорона праці, електробезпека та штучне освітлення	71
Висновки	74
Перелік джерел посилання	76
Додаток А Демонстраційний матеріал	80

ПЕРЕЛІК СКОРОЧЕНЬ

ПК – персонального комп'ютера;

САПР – системи автоматизованого проектування;

CCW – Counterclock (пропелер, який обертається проти годинникової стрілки, також називається лівим пропелером);

CW – Clockwise, (пропелер, що обертається за годинниковою стрілкою, також відомий як правий пропелер);

DoF – ступені свободи;

ENR – екологічний гібридний робот;

IMU – інерційний вимірювальний блок.

ВСТУП

Четверта промислова революція, або «індустрія 4.0», визначається використанням цифрових технологій, автоматизації та обміну даними у виробничих процесах. Щоб підвищити продуктивність, адаптивність і гнучкість у промисловому середовищі, розроблено інтелектуальні роботизовані системи. Інтеграція інтелектуальних робототехнічних систем у контексті «Індустрії 4.0» започаткувала нову еру розумного виробництва, трансформуючи традиційні виробничі процеси [1].

«Індустрія 4.0» спричинила зміну парадигми, у якій інтелектуальні роботизовані технології є важливими для зміни середовища промислових операцій. Впровадження Industry 4.0, четвертої промислової революції, що визначається автоматизованими, інтелектуальними та пов'язаними виробничими процесами, зробило інтелектуальні роботизовані системи незамінними [2].

Інтелектуальні роботизовані системи в Industry 4.0 характеризуються своєю адаптивністю до мінливих виробничих вимог. Завдяки вдосконаленим датчикам і алгоритмам навчання ці системи можуть динамічно коригувати свою поведінку, інструменти або програмування відповідно до варіацій у специфікаціях продукту або виробничих вимог [1]. Ця адаптивність сприяє розробці гнучких виробничих систем, здатних виконувати різноманітні завдання та виробляти продукти. Концепція співпраці «людина-машина» знаходиться в авангарді Industry 4.0, і інтелектуальні роботизовані системи відіграють передову роль у реалізації цього бачення. Ці системи сприяють безперервній взаємодії між людьми та машинами, сприяючи спільному робочому середовищу. Людський внесок стає невід'ємною частиною навчального процесу для роботів, підвищуючи загальну продуктивність і адаптивність [2].

Інтелектуальні роботизовані системи, оснащені вдосконаленими

системами бачення, сприяють контролю якості та контролю якості на виробництві. Ці системи забезпечують послідовний і точний контроль, що веде до покращення якості продукції та дотримання суворих стандартів якості [3]. Адаптивність інтелектуальних роботизованих систем в Industry 4.0 заслуговує похвали, дозволяючи безперебійне налаштування відповідно до конкретних потреб наших різноманітних виробничих завдань. Крім того, впровадження передової робототехніки в нашу промислову структуру оптимізувало робочі процеси, дозволяючи пришвидшити час виконання робіт і підвищити загальну продуктивність [3].

У розробці інтелектуальних робототехнічних платформ ключовим аспектом є організація системного обслуговування, яке охоплює не лише технічне обслуговування компонентів, а й підтримку програмного забезпечення, оновлення моделей машинного навчання та збереження цілісності даних.

Особливістю таких систем є можливість віддаленої діагностики, використання інтелектуальних агентів для моніторингу працездатності та впровадження стратегій профілактичного сервісу. Це забезпечує зниження простоїв, економію ресурсів та підвищення надійності платформи в цілому.

Мета роботи – розроблення програмної структури інтелектуальної системи керування мобільною роботизованою платформою з метою підвищення її адаптивності та ефективності за рахунок використання сенсорної інформації, нечіткого керування та візуалізації руху в середовищі моделювання.

Об'єкт розробки – інтелектуальне керування робототехнічною платформою.

Предмет розробки – робототехнічна платформа із інтелектуальним керуванням.

Робота виконано згідно [4-5]. Проведені дослідження відповідають цілям сталого розвитку (ЦСР) 9, так як наукові дослідження та інновації є важливими факторами, що сприяють сталому розвитку.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ЛІТЕРАТУРИ

1.1 Аналіз предметної області

У виробництві та логістиці автономні керовані транспортні засоби використовуються для обробки матеріалів і транспортування, підвищуючи ефективність складів і виробничих потужностей. Роботи, оснащені системами зору та штучним інтелектом, можуть автономно перевіряти та оцінювати якість продукції, виявляючи дефекти та забезпечуючи високі стандарти. Інтелектуальні роботизовані системи часто оснащені датчиками, які збирають і передають дані в реальному часі. Ці роботи можуть взаємодіяти з іншими пристроями та системами завдяки інтеграції Інтернету речей (IoT), що робить виробниче середовище гладким і мережевим [6]. Цифрові близнюки, віртуальні копії фізичних систем, дозволяють моделювати та тестувати роботизовані процеси перед їх впровадженням. Це зменшує ризик помилок і допомагає оптимізувати робочі процеси. Індустрія 4.0 спирається на кіберфізичні системи, де фізичні процеси контролюються та відстежуються в режимі реального часу.

Інтелектуальні роботизовані системи є невід’ємними компонентами цієї парадигми, сприяючи ефективності та гнучкості виробничих операцій. Інтелектуальні роботизовані системи сприяють автоматизації обробки матеріалів і логістики в ланцюжку постачання, що сприяє покращенню управління запасами та скороченню часу виконання [6]. Великі обсяги даних, створених під час промислових операцій, також можуть оброблятися та аналізуватися цими системами за допомогою аналітики даних. Прогнозне технічне обслуговування, оптимізація процесів і контроль якості стали можливими завдяки цій керованій даними методології.

Штучний інтелект (ШІ) допомагає передбачати несправності

обладнання та планувати завдання з технічного обслуговування до поломки, мінімізуючи час простою та підвищуючи загальну ефективність обладнання. Алгоритми штучного інтелекту та машинного навчання використовуються інтелектуальними роботизованими системами, щоб допомогти їм адаптуватися та вчитися на їхньому досвіді [7]. Ця адаптивність є цінною для оптимізації виробничих процесів і підвищення ефективності з часом [8].

Таким чином, інтеграція інтелектуальних роботизованих систем в Industry 4.0 трансформує традиційні методи виробництва, пропонуючи більшу гнучкість, ефективність і чутливість до динамічних вимог сучасного промислового середовища. Досягнення інтелектуальних робототехнічних систем і сенсорних технологій в Industry 4.0 показано на рис. 1.1.



Рисунок 1.1 – Досягнення інтелектуальних робототехнічних систем і сенсорних технологій в Industry 4.0 [8]

Алгоритми штучного інтелекту та машинного навчання дозволяють роботам навчатися на даних, адаптуватися до мінливих умов і з часом

покращувати свою продуктивність. Ця можливість адаптивного навчання необхідна для роботи в динамічних і непередбачуваних виробничих середовищах. Інтелектуальні роботизовані системи розроблені для оптимізації енергоспоживання та використання ресурсів.

1.1.1 Підключена автоматизація

Інтелектуальні роботизовані системи відіграють центральну роль в «Індустрії 4.0», будучи плавно пов'язаними з іншими компонентами виробничої екосистеми [6]. Вони взаємодіють із датчиками, машинами та іншими роботами, щоб створити високо взаємопов'язане та автоматизоване виробниче середовище [7]. Інтелектуальні роботизовані системи служать основою цього зв'язку, сприяючи безперебійному зв'язку між машинами, датчиками та широким виробництвом. екосистема [8]. Ця взаємопов'язана автоматизація підвищує загальну операційну ефективність і оперативність. У цьому контексті інтелектуальні роботизовані системи використовують підключену автоматизацію для підвищення ефективності, гнучкості та швидкості реагування в промислових процесах.

Ось деякі ключові аспекти підключеної автоматизації в інтелектуальних робототехнічних системах в Industry 4.0 [9]:

1) заємозв'язані пристрої та системи. Інтелектуальні роботизовані системи в Industry 4.0 характеризуються заємозв'язністю пристроїв і систем. Роботи, датчики, приводи та інше виробниче обладнання підключено до мережі, що забезпечує безперебійний зв'язок і обмін інформацією;

2) промисловий Інтернет речей (ІІоТ): мережева автоматизація базується на ІІоТ. Дані в режимі реального часу збираються за допомогою датчиків, вбудованих у виробниче обладнання, і роботизованих систем, які пропонують уявлення про роботу та стан машин. Потім ці дані використовуються для прийняття рішень та оптимізації;

3) аналітика даних і машинне навчання. Дані, створені підключеними робототехнічними системами, аналізуються за допомогою розширеної

аналітики та алгоритмів машинного навчання. Це дозволяє здійснювати прогнозне технічне обслуговування, оптимізувати процеси та ідентифікувати шаблони, які можуть підвищити загальну ефективність і скоротити простої;

4) роботи для співпраці (коботи): підключена автоматизація полегшує розгортання роботів для співпраці або коботів. Ці роботи можуть працювати разом з людьми-операторами, а їхні дії можуть бути синхронізовані з іншими машинами на виробничій лінії. Цей спільний підхід підвищує гнучкість і адаптивність у виробничих процесах;

5) моніторинг і управління в режимі реального часу: завдяки підключеній автоматизації оператори можуть контролювати та контролювати роботизовані системи в режимі реального часу. Ця можливість необхідна для швидкого коригування виробничих процесів, оперативного вирішення проблем та оптимізації використання ресурсів;

6) віддалене технічне обслуговування та діагностика: підключена автоматизація дозволяє здійснювати дистанційний моніторинг і діагностику робототехнічних систем. Проблеми з техобслуговуванням можна виявити на ранній стадії, а в деяких випадках проблеми можна вирішити дистанційно, зменшуючи потребу у фізичному втручанні та мінімізуючи час простою;

7) заходи кібербезпеки: враховуючи розширене підключення, кібербезпека є критично важливим питанням. Впровадження Індустрії 4.0 повинні включати надійні заходи кібербезпеки для захисту від потенційних загроз і забезпечення цілісності даних і операцій;

8) цифрові двійники: цифрові близнюки — це цифрові копії процесів або систем реального світу. Цифрові двійники можна використовувати для тестування, моделювання та оптимізації в контексті інтелектуальних роботизованих систем, що дозволяє краще зрозуміти поведінку та функціональність системи;

9) інтеграція ланцюга поставок: підключена автоматизація виходить за межі окремих фабрик і інтегрується в ширший ланцюг постачання. Ця інтеграція забезпечує кращу координацію та синхронізацію виробничих

процесів на різних етапах виробництва;

10) адаптивне та гнучке виробництво: підключена автоматизація в інтелектуальних роботизованих системах підтримує адаптивні та гнучкі виробничі процеси. Системи можуть швидко адаптуватися до змін попиту, переконфігурувати виробничі лінії та оптимізувати використання ресурсів на основі даних у реальному часі.

Модель роботи хмарного виробництва показана на рис. 1.2.

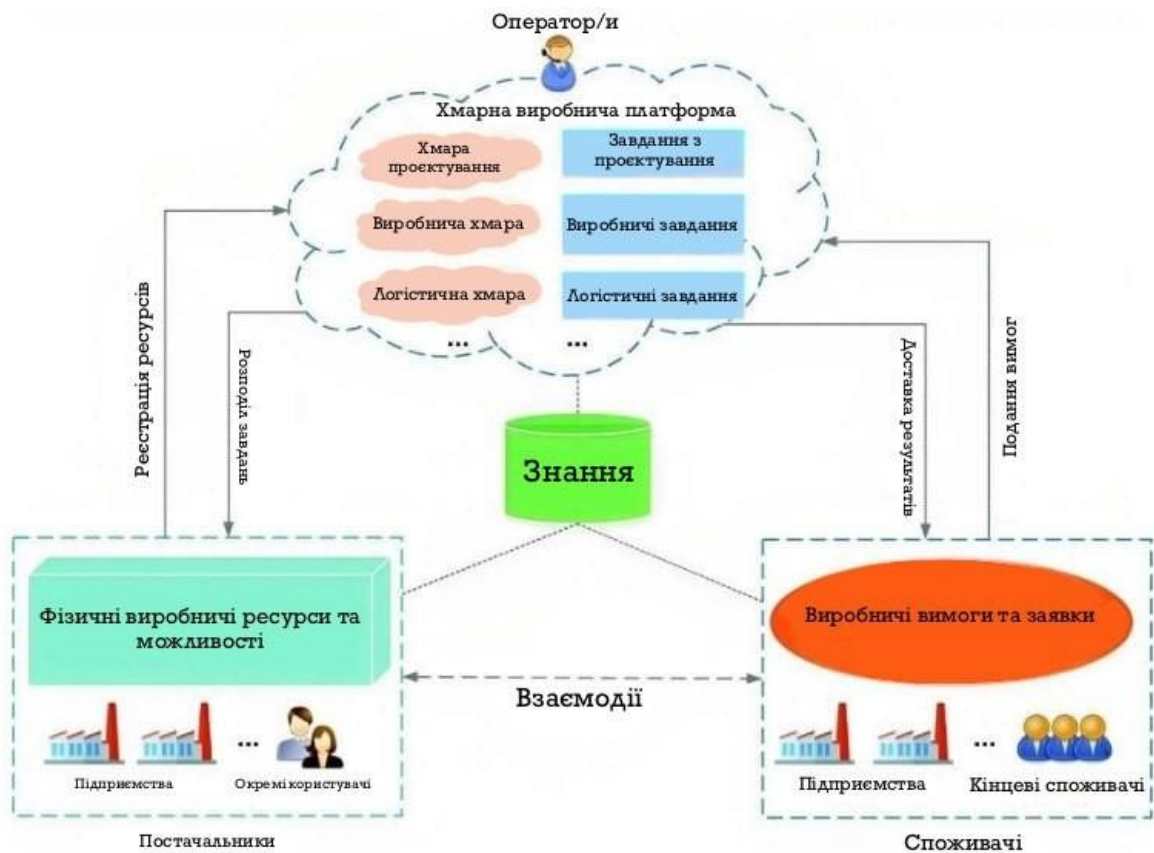


Рисунок 1.2 – Модель роботи хмарного виробництва [10]

Підсумовуючи, підключена автоматизація є фундаментальним аспектом інтелектуальних робототехнічних систем в Industry 4.0, що сприяє підвищенню ефективності, гнучкості та швидкості реагування у виробничих процесах. Інтеграція цифрових технологій і підключення не тільки розширює можливості окремих робототехнічних систем, але й забезпечує більш цілісний і взаємопов'язаний підхід до промислової автоматизації.

1.1.2 Автономна навігація

Індустрія 4.0 охоплює інтелектуальні роботизовані системи з можливостями автономної навігації. Ці роботи можуть переміщатися в динамічному середовищі, уникаючи перешкод і оптимізуючи свій шлях для ефективної обробки матеріалів і логістики на розумних фабриках [11]. Ось ключові аспекти, пов'язані з автономною навігацією в інтелектуальних роботизованих системах в Індустрії 4.0:

1) датчики та сприйняття Щоб відчувати оточення, інтелектуальні роботизовані системи використовують різні датчики, зокрема інерціальні, радарні, LiDAR і камери. Дані кількох датчиків об'єднуються за допомогою методів об'єднання датчиків, щоб отримати повну картину навколишнього середовища. Алгоритми комп'ютерного зору дозволяють роботам інтерпретувати візуальну інформацію, розпізнавати об'єкти та здійснювати навігацію на основі аналізу зображень або відеоканалів;

2) алгоритми відображення та локалізації: роботи можуть відображати на карті своє оточення та визначати своє власне розташування всередині нього одночасно завдяки процедурам локалізації та відображення одночасно. Це важливо для навігації в динамічних середовищах. Розширені алгоритми локалізації, такі як локалізація Монте-Карло (MCL) або фільтрація Калмана, використовуються для точної оцінки положення робота;

3) алгоритми планування шляху та прийняття рішень: ці алгоритми допомагають роботам визначати найбільш ефективний маршрут від однієї точки до іншої, уникаючи перешкод. В результаті роботи можуть рухатися за робочим графіком, уникаючи перешкод і оптимізуючи свій шлях під час робочих умов;

4) алгоритми уникнення перешкод і виявлення зіткнень: роботам потрібна здатність динамічно адаптувати свій шлях, щоб уникнути перешкод. Це передбачає прийняття рішень у реальному часі на основі даних датчиків.

5) комунікація та співпраця за допомогою взаємопов'язаних систем: в Індустрії 4.0 роботизованим системам часто потрібно співпрацювати з іншими

машинами та системами. Комунікаційні протоколи, такі як OPC UA (уніфікована архітектура комунікацій відкритої платформи), сприяють безперебійному обміну інформацією між різними компонентами виробничої екосистеми;

6) машинне навчання та штучний інтелект за допомогою навчання з підкріпленням: методи штучного інтелекту, такі як навчання з підкріпленням, можна використовувати, щоб дозволити роботам вивчати оптимальні стратегії навігації в складних і динамічних середовищах. Алгоритми, навчені на попередніх даних, можуть прогнозувати можливі перешкоди, несправності в обладнанні та інші елементи, які можуть вплинути на навігацію;

7) безпека та дотримання нормативних вимог шляхом уникнення зіткнень: впровадження заходів безпеки, таких як системи виявлення та уникнення зіткнень, має вирішальне значення для запобігання нещасним випадкам і пошкодженням роботизованої системи та її оточення;

8) відповідність нормам. Робототехнічні системи повинні відповідати галузевим нормам і нормам безпеки, які регулюють автономну навігацію в промислових умовах, щоб підвищити безпеку робочих умов.

9) моніторинг у режимі реального часу та можливість адаптації за допомогою інтеграції Інтернету речей: технології Інтернету речей (IoT) дозволяють здійснювати моніторинг робототехнічних систем у режимі реального часу. Ці дані можна використовувати для прийняття рішень на ходу та адаптації навігаційних стратегій на основі мінливих умов;

10) енергоефективність завдяки оптимізованим маршрутам: автономні навігаційні системи можуть оптимізувати маршрути не лише для економії часу, але й для економії енергії, сприяючи загальній стійкості у виробництві.

Додатково до традиційних підходів, у практиці інтелектуального планування траєкторій активно використовуються алгоритми на основі графів, зокрема Dijkstra, A, Theta та їх модифікації. Ці методи дозволяють побудувати найкоротший або найнадійніший маршрут у середовищі з перешкодами на основі топології карти [12].

Застосування методів планування шляху в реальному часі, таких як RRT (Rapidly-exploring Random Tree), дає змогу ефективно будувати траєкторії в умовах обмеженого часу та змінного середовища.

Методи потенційних полів забезпечують швидке ухилення від перешкод завдяки моделюванню штучних сил, однак потребують комбінування з іншими алгоритмами для уникнення локальних мінімумів.

Рисунок 1.3 ілюструє інтелектуальне виробництво з використанням технології AGV, інтегрованої у виробництво модулів [12].

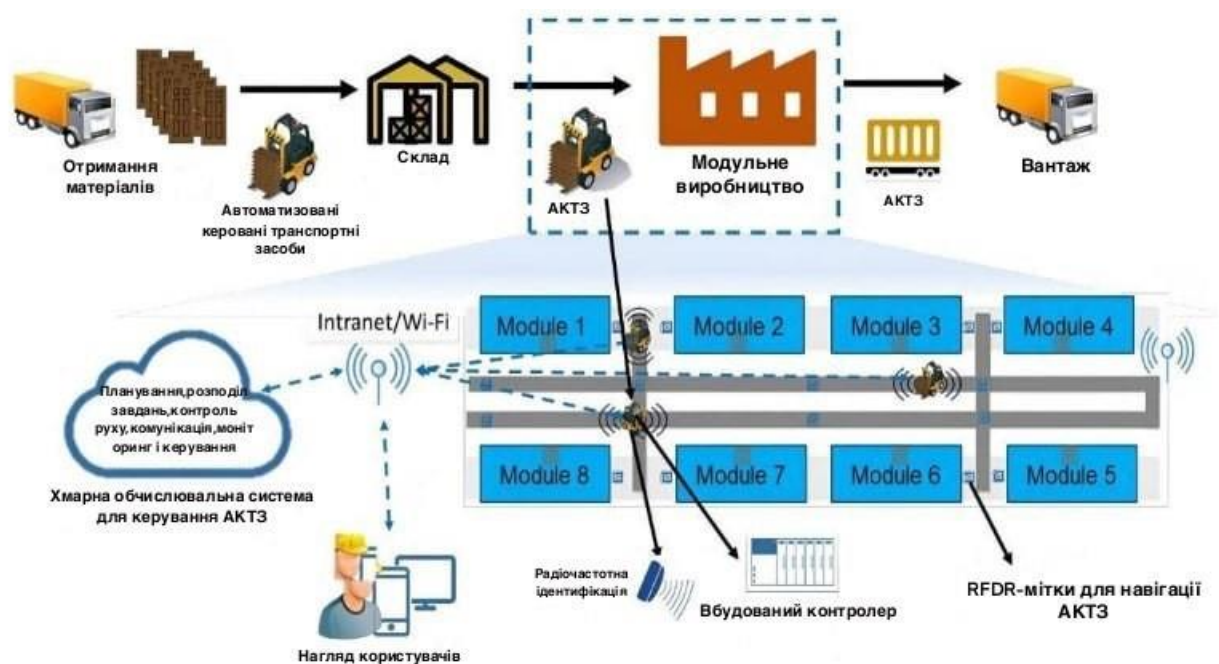


Рисунок 1.3 – Розумне виробництво з використанням технології AGV, інтегрованої у виробництво модулів [12]

1.1.3 Співпраця людини і машини

Перехід «Індустрії 4.0» до співпраці людини та робота найкраще демонструє поява роботів, що співпрацюють, або коботів. Ці роботи працюють разом з людьми-операторами, підвищуючи продуктивність, безпеку та гнучкість. Інтелектуальні роботизовані системи сприяють безперервній співпраці між людьми та машинами. Працівники-люди можуть взаємодіяти з роботами через інтуїтивно зрозумілі інтерфейси, а роботи

можуть вчитися на основі інформації людини, створюючи синергетичне та ефективне робоче середовище. Синергія між людьми та роботами створює гармонійне робоче середовище, де кожен доповнює сильні сторони іншого. Ось ключові аспекти співпраці людини і машини в інтелектуальних робототехнічних системах в рамках Індустрії 4.0:

1) коботи (колаборативні роботи): колаборативні роботи розроблені для роботи разом із працівниками в спільному робочому просторі. Ці роботи оснащені передовими датчиками та програмуванням для забезпечення безпечної та ефективної співпраці. Вони можуть виконувати завдання, які або надто небезпечні, або монотонні для людей, дозволяючи робочій силі зосередитися на більш складних і творчих аспектах своєї роботи. Робочі процедури для співпраці згідно з ISO 10218-1/2 показані на рисунку 1.4 [13];

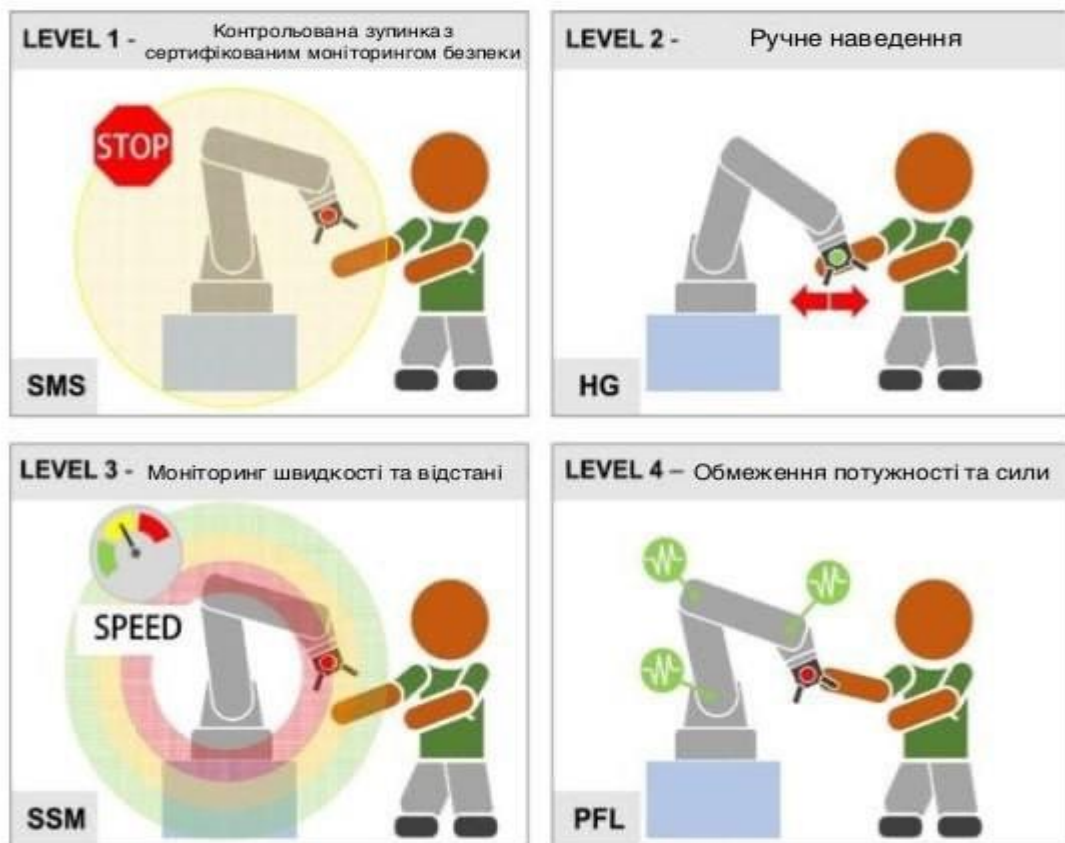


Рисунок 1.4 – Робочі процедури для співпраці згідно з ISO 10218-1/2 [13]

2) підвищення навичок: інтелектуальні роботизовані системи можуть розширити можливості людини, виконуючи повторювані та фізично складні завдання. Ця співпраця підвищує загальну продуктивність і якість, одночасно знижуючи ризик людської помилки. Працівники можуть зосередитися на завданнях, які вимагають критичного мислення, вирішення проблем і креативності;

3) системи людини в циклі: Індустрія 4.0 наголошує на даних у реальному часі та підключенні. Працівники можуть брати активну участь у процесах прийняття рішень за допомогою систем «людина в циклі». Інтелектуальні роботизовані системи збирають і аналізують дані, а люди забезпечують розуміння контексту, приймають складні рішення та вирішують нестандартні ситуації;

4) адаптивна автоматизація: інтелектуальні роботизовані системи в Industry 4.0 створені для адаптації. Вони можуть вчитися у людей-операторів, адаптуватися до змін у навколишньому середовищі та постійно покращувати свою продуктивність. Ця адаптивність має вирішальне значення в динамічних виробничих середовищах, де процеси та вимоги можуть часто змінюватися;

5) зручні інтерфейси: співпраця між людьми та машинами полегшується завдяки зручним інтерфейсам. Люди-оператори повинні безперебійно взаємодіяти з інтелектуальними робототехнічними системами. Інтуїтивно зрозумілі інтерфейси, доповнена реальність і обробка природної мови сприяють ефективному спілкуванню між людьми та машинами;

6) обмін даними та інтеграція: у середовищі розумної фабрики дані є ключовим активом. Співпраця людини і машини передбачає обмін та інтеграцію даних з різних джерел. Цей обмін даними дозволяє краще приймати рішення, прогнозувати технічне обслуговування та оптимізувати виробничі процеси;

7) безперервне навчання та освіта: у міру розвитку технологій безперервне навчання та освіта стають важливими як для працівників, так і для інтелектуальних робототехнічних систем. Працівники повинні бути знайомі з

роботою та програмуванням цих систем, а самі системи потребують оновлень і вдосконалень, щоб залишатися актуальними;

8) етичні міркування та безпека. Співпраця людини та машини має бути розроблена з урахуванням етичних міркувань. Необхідно вжити заходів безпеки, щоб захистити працівників, а етичні міркування, такі як переміщення з роботи, мають бути розглянуті шляхом продуманого проектування та реалізації.

На рис. 1.5 зображено самокеровану структуру для мобільних роботів, які є пристроями, оснащеними функціями захисту людини.

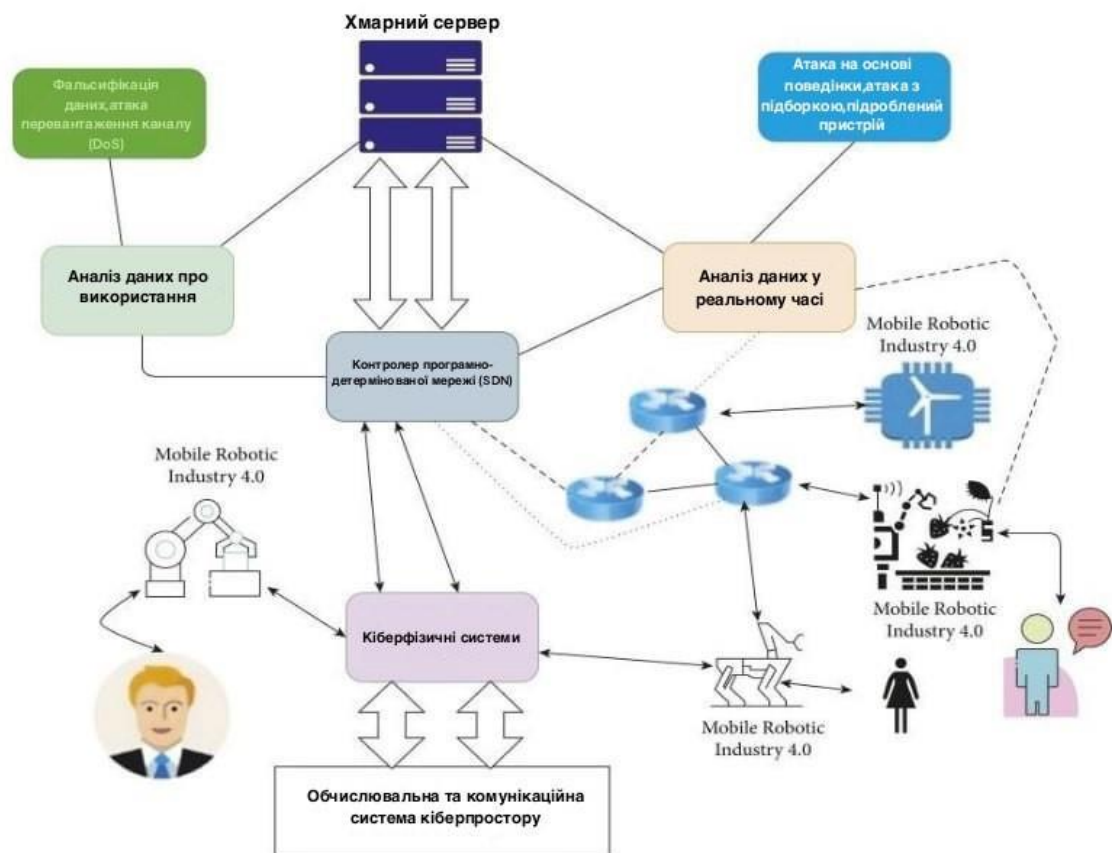


Рисунок 1.5 – Самокерована структура для мобільних роботів, які є пристроями, оснащеними функціями безпеки людини [14]

У межах дослідження також розглядається структура ієрархічних освітніх систем, що мають вагоме значення для формування інтелектуальних алгоритмів тестування та діагностики. Ієрархічна модель передбачає

декомпозицію процесу навчання або контролю знань на кілька рівнів – від базових перевірок до аналітичного оцінювання й адаптивного коригування стратегії навчання.

На нижчих рівнях функціонують прості алгоритми перевірки правильності виконання завдань (true/false), а вищі рівні включають аналіз помилок, побудову профілю користувача, коригування інтерфейсу взаємодії та надання індивідуальних рекомендацій. Такий підхід дає змогу гнучко адаптуватися до рівня підготовки оператора або користувача інтелектуальної системи [14].

Співпраця між людьми та інтелектуальними роботизованими системами в Industry 4.0 є динамічною сферою, що розвивається. Він має потенціал революціонізувати виробничі процеси, підвищити ефективність і створити безпечніші та гнучкіші робочі середовища. Оскільки технології продовжують розвиватися, головне – знайти правильний баланс між автоматизацією та втручанням людини, забезпечуючи гармонійну та продуктивну співпрацю.

1.1.4 Гнучкі виробничі системи

У контексті інтелектуальних роботизованих систем Industry 4.0 гнучкі виробничі системи (FMS) є важливими. Індустрія 4.0 відома своїми гнучкими виробничими процесами, які значно вдосконалені інтелектуальними роботизованими системами. Інтелектуальні роботи допомагають створювати гнучкі виробничі системи, які можуть швидко пристосовуватися до мінливих вимог до виробництва. Ці системи можуть пристосовуватися до змін у виробничих вимогах, забезпечуючи швидку реконфігурацію та ефективне виконання різноманітних завдань, від дрібносерійного виробництва до індивідуального виробництва. Ці системи чудово підходять для дрібносерійного або індивідуального виробництва, підтримуючи тенденцію до персоналізованого та індивідуального виробництва. виробництво за вимогою. Ось як гнучкі виробничі системи інтегруються в інтелектуальні роботизовані системи в парадигмі Індустрії 4.0:

1) взаємозв'язані системи: FMS в Industry 4.0 характеризується взаємопов'язаними системами, де різні компоненти, включаючи роботизовані системи, датчики та машини, безперешкодно взаємодіють через Інтернет речей (IoT) та інші комунікаційні протоколи. Робототехніка в FMS може отримувати дані в режимі реального часу від інших підключених пристроїв, що дозволяє їм адаптуватися до мінливих умов, оптимізувати графіки виробництва та приймати обґрунтовані рішення;

2) адаптивна автоматизація: інтелектуальні роботизовані системи в Industry 4.0 розроблені як адаптивні, здатні коригувати свою поведінку на основі даних у реальному часі та зворотного зв'язку. FMS сприяє цій адаптивності, дозволяючи роботам перемикатися між завданнями, змінювати конфігурацію виробничих ліній і пристосовуватися до коливань попиту чи доступності ресурсів;

3) спільна робототехніка: FMS часто включають спільних роботів (коботів), які можуть працювати разом з людьми-операторами. Ці роботи оснащені передовими датчиками та системами зору, що дозволяє їм безпечно взаємодіяти з людьми. Спільні роботи підвищують гнучкість у виробництві, легко адаптуючись до нових завдань і працюючи в безпосередній близькості від людей без шкоди для безпеки;

4) прийняття рішень на основі даних: FMS генерує величезну кількість даних від датчиків, машин і роботів. Інтелектуальні роботизовані системи використовують ці дані для прогнозного обслуговування, оптимізації процесів і контролю якості. Розширена аналітика, машинне навчання та алгоритми штучного інтелекту аналізують ці дані, щоб приймати обґрунтовані рішення, виявляти закономірності та постійно вдосконалювати виробничі процеси;

5) моніторинг і контроль у реальному часі: FMS в Industry 4.0 забезпечує моніторинг і контроль виробничих процесів у реальному часі. Робототехніку в цих системах можна контролювати дистанційно, і коригування можна вносити в режимі реального часу для оптимізації продуктивності, скорочення часу простою та підвищення загальної ефективності;

б) модульні та масштабовані системи: FMS часто розроблені як модульні та масштабовані, що дозволяє легко інтегрувати нові роботизовані технології або реконфігурувати існуючі системи відповідно до мінливих вимог виробництва. Динамічний характер Industry 4.0 відповідає підвищеній адаптивності та гнучкості виробничої системи завдяки її модульній конструкції;

7) співпраця людини та машини: FMS сприяє ефективній співпраці між людьми та роботами. Інтелектуальні робототехнічні системи розроблені таким чином, щоб розуміти та реагувати на вхід людини, що полегшує для працівників взаємодію з роботизованими процесами та контроль за ними.

Гнучка виробнича система з використанням інтелектуальних роботизованих систем показана на рис. 1.6.

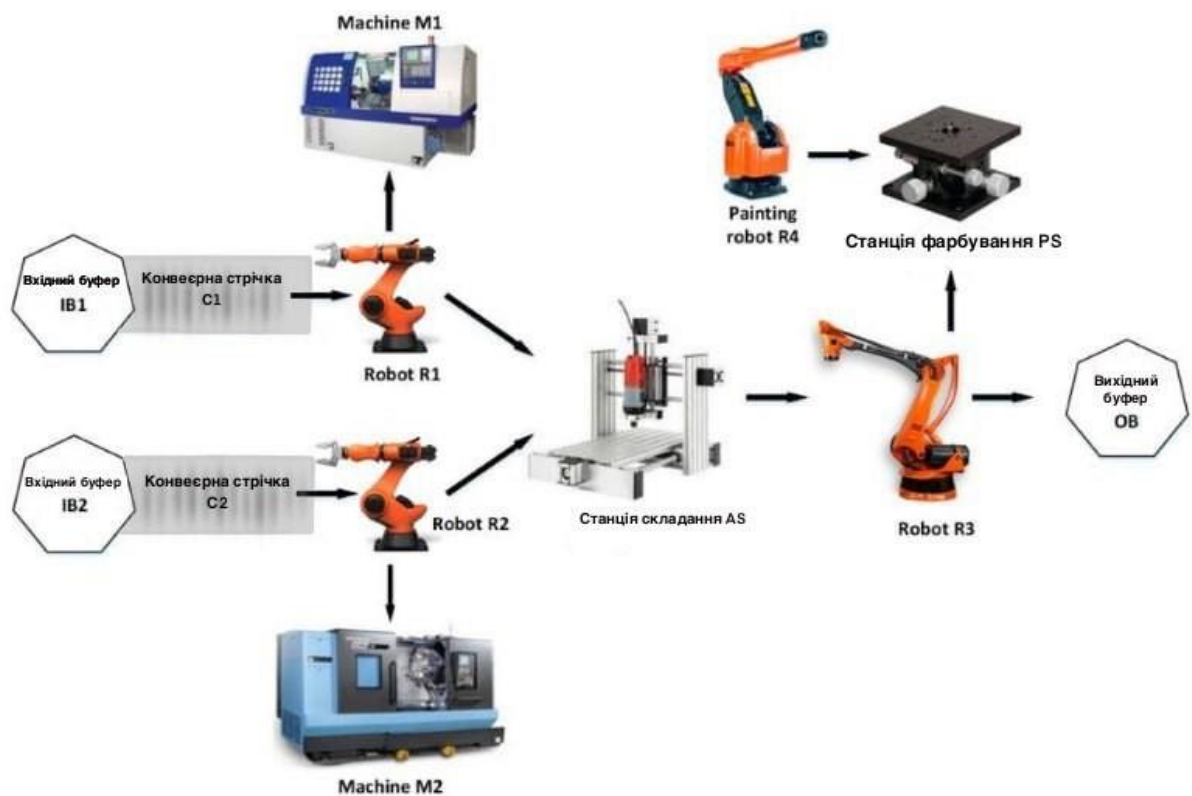


Рисунок 1.6 – Гнучка виробнича система з використанням інтелектуальних роботизованих систем [14]

Таким чином, інтеграція гнучких виробничих систем з інтелектуальними роботизованими системами в Industry 4.0 призводить до створення адаптивних, керованих даними та взаємопов'язаних виробничих середовищ, які можуть швидко реагувати на зміни умов, оптимізувати ефективність і підвищити загальну продуктивність.

1.1.5 Сучасні складальні лінії Intelligent Robotic Systems

У контексті Індустрії 4.0 складальні лінії в інтелектуальних роботизованих системах передбачають інтеграцію передових технологій для забезпечення більш адаптивних, ефективних і інтелектуальних виробничих процесів.

Таким чином, складальні лінії в контексті Industry 4.0 використовують поєднання робототехніки, Інтернету речей, аналітики, ШІ та інших передових технологій для створення більш ефективних, гнучких і інтелектуальних виробничих процесів. Основні моменти, на які приділяється увага, – співпраця людини і машини, прийняття рішень на основі даних і гнучкість.

Гібридні системи обслуговування поєднують традиційне планове технічне обслуговування з прогнозними механізмами, які ґрунтуються на даних сенсорів та аналітиці великих даних. Такий підхід дозволяє формувати динамічні графіки обслуговування з урахуванням фактичного стану обладнання.

Застосування машинного навчання в аналізі параметрів функціонування обладнання дає змогу виявляти закономірності, що передують зниженню ефективності або відмовам. Таким чином, гібридна система забезпечує найкращий баланс між витратами на сервіс та безперервністю роботи робототехнічної платформи.

1.1.6 Контроль та перевірка якості

Контроль якості та перевірка відіграють вирішальну роль у забезпеченні надійності та ефективності інтелектуальних роботизованих систем у контексті

Індустрії 4.0. Роботизовані системи, оснащені інтелектуальними системами бачення, забезпечують високу якість виробництва, виконуючи точний і послідовний контроль якості та перевірку. Вони можуть виявляти дефекти, вимірювати розміри та забезпечувати відповідність стандартам якості.

Моніторинг здоров'я виробничого середовища за допомогою ШІ показано на рис. 1.7 [15].



Рисунок 1.7 – Моніторинг стану виробничого середовища за допомогою ШІ [15]

1.1.7 Прийняття рішень на основі даних

Індустрія 4.0 відрізняється тим, що вона значною мірою покладається на прийняття рішень на основі даних, а інтелектуальні роботизовані системи є важливими для цієї парадигми. Великі обсяги даних збираються, обробляються та аналізуються в режимі реального часу цими системами завдяки складним датчикам і аналітиці. Статистика, отримана з цих даних, дає змогу приймати проактивні рішення, прогнозувати технічне обслуговування та безперервну оптимізацію процесів.

Серед фреймворків, які використовуються для створення інтелектуальних систем керування, найпоширенішим є ROS (Robot Operating

System), що забезпечує обмін повідомленнями, керування процесами та інтеграцію модулів. Для візуального аналізу використовуються OpenCV, а для реалізації ШІ – TensorFlow, Keras, PyTorch.

Для симуляції та моделювання фізичних взаємодій платформи застосовуються Gazebo, Webots, CoppeliaSim. Ці середовища дозволяють проводити тестування алгоритмів навігації, зору, маніпулювання та взаємодії без необхідності мати фізичний прототип.

На рис. 1.8 показано дослідження інтелектуального прийняття рішень із застосуванням промислових технологій, що базуються на великих даних [16].



Рисунок 1.8 – Інтелектуальне прийняття рішень із застосуванням промислової технології, що базується на великих даних [16]

Таким чином, прийняття рішень на основі даних в інтелектуальних робототехнічних системах в рамках Industry 4.0 має важливе значення для досягнення ефективності, якості, адаптивності та загальної операційної досконалості у виробничих процесах. Аналіз даних у реальному часі, створених робототехнічними системами, дає змогу виробникам приймати обґрунтовані рішення, які стимулюють постійне вдосконалення та інновації в мінливому ландшафті промислових автоматичних систем.

1.1.8 Енергоефективність в інтелектуальних робототехнічних системах

Індустрія 4.0 наголошує на стійкості, а інтелектуальні роботизовані системи сприяють енергоефективному виробництву, оптимізуючи їхні рухи, скорочуючи час простою та впроваджуючи функції енергозбереження. Інтелектуальні роботизовані системи в Industry 4.0 надають пріоритет енергоефективності та сталості. Оптимізація енергоспоживання в роботизованих системах не лише узгоджується з цілями сталого розвитку, але також сприяє економії коштів і загальній ефективності роботи. Завдяки оптимізованим рухам, скороченню часу простою та функціям енергозбереження ці системи сприяють досягненню головної мети створення екологічно свідомих виробничих практик.

Інтеграція цих міркувань у проектування, розробку та експлуатацію інтелектуальних робототехнічних систем може значно сприяти досягненню цілей енергоефективності в умовах промисловості 4.0 важливо прийняти цілісний підхід, який враховує весь життєвий цикл роботизованої системи та збалансовує енергоефективність із вимогами до продуктивності.

1.1.9 Робота інтелектуальних систем у різних середовищах

Ефективність інтелектуальних систем залежить від середовища, у якому вони працюють. У статичному середовищі алгоритми працюють за стабільними параметрами, що дозволяє зменшити обчислювальні витрати. У динамічному середовищі платформи повинні враховувати змінні об'єкти, адаптувати навігацію та застосовувати SLAM, щоб формувати карту і визначати власне положення.

У несприятливих умовах – при пилу, вологості, температурних коливаннях – використовуються герметичні корпуси, компенсація шуму в сенсорах, а також програмні фільтри для очищення сигналів.

У взаємодії з людьми необхідне дотримання стандартів безпеки, використання інтуїтивних інтерфейсів, контролю жестів або мови, що робить коботів ефективними у спільному просторі.

1.2 Аналіз технічного завдання

1.2.1 Аналіз літератури

Розробка методів і засобів управління мобільними роботизованими системами з використанням обчислювального інтелекту висвітлена в працях вчених, а саме: Адіна Анікулесей та інші – розробка системи управління інтелектуальною мобільною робототехнічною системою з самонавчанням;

Однак використання МРП спеціального призначення вимагає високого рівня автономності системи, який пропонується реалізувати з комплексним підходом за рахунок використання методів обчислювального інтелекту. Роботи дають часткові рішення по впровадженню інтелектуальної системи управління МРП. У випадку з мобільним роботом його система зв'язку повинна мати високий рівень стабільності та захисту даних.

Звичайні датчики мають обмеження з точки зору дальності виявлення, просторової роздільної здатності та складності обробки. Однак перспективним підходом є оснащення автономного мобільного робота датчиком виявлення світла та дальності (LiDAR) для уникнення перешкод. В [17] представлені всеспрямовані мобільні платформи, які можуть самостійно і одночасно виконувати поступальні і обертальні рухи. З метою розробки автономного всеспрямованого мобільного маніпулятора представлена платформа на основі чотирьох коліс Mecanum, що володіють більш високою вантажопідйомністю і мобільністю, ніж стандартна чотириколісна платформа.

Робота [18] присвячена автономній навігації мобільного робота з поточного положення в потрібне тільки з використанням поточного візуального спостереження, без попередньо складеної карти навколишнього середовища. В рамках навчання з глибоким підкріпленням використовується мережа Deep Q Network (DQN) для досягнення відображення від вихідного зображення до оптимальної дії мобільного робота.

У [19] представлена автономна навігаційна система для внутрішніх мобільних роботів в основному на основі відкритого вихідного коду, що

забезпечується операційною системою робота. Представлена система здатна автономно переміщатися по неструктурованому внутрішньому середовищі, уникаючи зіткнення зі статичними або динамічними об'єктами.

Ще одним викликом є побудова системи управління роботизованою платформою. У роботі [20] описується застосування методів нечіткої логіки в управлінні швидкістю мобільного робота шестиногого за допомогою типу Такагі-Сугено-Канга в його системі висновків. У статті [21] описується проектування та реалізація колісного мобільного робота з використанням принципів нечіткої логіки з системою нечіткого висновку Мамдані, щоб робот мав здатність уникати перешкод.

Перспективним підходом є поєднання нейронних мереж і методів нечіткої логіки, як описано в [22], для досягнення позитивних результатів. З метою вирішення проблеми слабкого ефекту керування траєкторією традиційних роботів був вивчений інтелектуальний алгоритм нечіткого керування мобільним траєкторією руху мобільних роботів. У цьому випадку інтелектуальне управління мобільним траєкторією робота здійснюється шляхом об'єднання нечіткого алгоритму, а параметри відхилення шляху і корекції відхилень розраховуються для завершення інтелектуального управління мобільним траєкторією робота. У статті [23] напівавтоматичний навігаційний робот використовує штучний інтелект Fuzzy logic як вихідний процесор, який буде генеруватися роботом. Нечітка логіка на цьому роботі використовується для управління швидкістю двигуна на основі відстані перешкоди, яка зчитується датчиком і входом, що забезпечується пультом дистанційного керування. У роботі [24] представлений метод нечіткого контролера для управління роботом для обходу всіх нерухомих і рухливих перешкод за допомогою нечіткої логіки. Роботом можна керувати за допомогою людино-машинного інтерфейсу або за допомогою клавіатури, а зв'язок між людино-машинним інтерфейсом і роботом здійснюється за допомогою бездротового зв'язку. Контролер нечіткої логіки, описаний в [25], використовується і оптимізується двома методами м'якого комп'ютера:

генетичним алгоритмом і оптимізацією рою частинок (PSO). Ці методи використовуються для регулювання входів і виходів контролера нечіткої логіки з метою поліпшення навігації мобільного робота і були представлені три з методів.

1.2.2 Аналіз технічного завдання

Створення МРП з високими техніко-експлуатаційними характеристиками вимагає розробки нових методів, алгоритмів і апаратних структур, орієнтованих на ефективну реалізацію алгоритмів обробки і розпізнавання, моделювання навколишнього середовища, планування дій, прокладання раціональних маршрутів руху і управління рухом. При управлінні рухом МРП пропонується використовувати нечітке логічне управління.

Основними вимогами до побудови ІСУР є забезпечення наступного:

- зменшення габаритів, енергоспоживання та вартості ІСУР;
- управління пересуванням в режимі реального часу;
- оптимізація маршруту за критеріями часу та відстані;
- збереження працездатності ІСУР в умовах впливу зовнішніх факторів;
- висока надійність обладнання та програмного забезпечення;
- можливість адаптації під вимоги замовника;
- універсальність, сумісність та інтеграція з існуючими МРП;
- здатність самостійно виконувати поставлені завдання в умовах невизначеності зовнішньої обстановки;
- використання ІСУР і ніч в умовах радіоелектронної та інформаційної протидії. Створення таких ІСУР вимагає широкого використання сучасних комплектуючих (датчиків, платформ, мікроконтролерів, SoCs, FPGA і т.д.), розробка нових методів, алгоритмів і засобів обробки в реальному часі.

Режим реального часу накладає обмеження на час обробки і формування команд управління T_{ce} , які не повинні перевищувати час накопичення T_{dl} даних, тобто:

$$T_{cc} \leq T_{dl} \quad (2.1)$$

Час накопичення масиву даних залежить як від його обсягу N , розіта даних N і частоти F_d їх прийому, так і від біта n_k і числа каналів k , які вони приймають. Цей час визначається за формулою:

$$T_{dl} = \frac{N_n}{F_d k n_k}. \quad (2.2)$$

Для забезпечення обробки і формування команд управління в реальному часі виконанням роботи P_c .

Комп'ютерні інструменти повинні бути:

$$p_c \geq \frac{\beta R F_d k n_k}{N_n}, \quad (2.3)$$

де R – складність алгоритмів обробки і формування управлінських команд;
 β – коефіцієнт врахування особливостей комп'ютерної реалізації.

Для забезпечення необхідної продуктивності комп'ютерних засобів для обробки потоків даних і формування команд управління в реальному часі необхідно використовувати спеціалізовані апаратні засоби, шкалювання і паралельну обробку даних.

При розробці ІСУР необхідно в першу чергу забезпечити режим реального часу і при мінімізації витрат на обладнання. Для оцінки ресурсів ІСУР пропонується використовувати

Критерій ефективності, який прив'язує час обробки даних до витрат на обладнання та дає оцінку продуктивності елементів системи.

Кількісно визначається величина ефективності використання обладнання в ІСУР:

$$E_{AC} \geq \frac{R_{AC}}{t_{AWH}}, \quad (2.4)$$

де R_{AC} – складність алгоритмів обробки потоків даних і команд управління;
 t_A – час виконання алгоритмів;
 W_H – витрати на обладнання для ІСУР.

Висока ефективність використання апаратури при розробці ІСУР пропонується досягати за рахунок використання універсальних ядер процесорів, доповнених спеціалізованими апаратними засобами, що реалізують складні обчислювальні алгоритми.

Вихідною інформацією для розвитку МРЦ є:

- алгоритми обробки потоків даних;
- методи та алгоритми нечіткого керування рухом МРП;
- база даних правил нечіткого управління рухом МРП;
- алгоритми фазифікації, дефазифікації та управлінські рішення;
- величина вхідного масиву N ;
- інтенсивність вхідних потоків;
- вимоги до зовнішніх інтерфейсів ІСУР;
- розрядність вхідних даних і необхідна точність розрахунків;
- технічні та експлуатаційні вимоги та обмеження.

У загальному завданні розробки ІСУР можна сформулювати так:

- обирати алгоритми обробки, фазифікації, розфазифікації, управлінських рішень та представляти їх у вигляді заданого потокового графіка;
- визначити порядок реалізації алгоритмів функціонування ІСУР;
- розробляти структури компонентів ІСУР з максимальною ефективністю використання обладнання, що враховують всі обмеження і забезпечують контроль руху МРП в режимі реального часу;
- визначити основні характеристики компонентів ІСУР та розробити їх;
- обирати способи обміну, виявляти необхідні зв'язки, розробляти систему обміну даними між компонентами ІСУР.

Розробку ІСУР пропонується здійснювати на основі комплексного підходу, який охоплює:

- методи навігації, методи попередньої обробки даних;
- сучасні методи та алгоритми інтелектуального управління, нечітка логіка;
- сучасна елементна база;
- методи інтелектуальної обробки та оцінки даних з датчиків в умовах перешкод і неповноти інформації;
- методи та засоби автоматизованого проектування апаратних і програмних засобів.

Реалізацію ІСУР пропонується виконувати на основі проблемно-орієнтованого підходу, який передбачає поєднання універсальних і спеціалізованих засобів, які важко реалізують найскладніші обчислювальні алгоритми.

Розробку ІСУР пропонується здійснювати на таких принципах:

- ієрархія побудови шляхом поділу її на рівні управління;
- систематичність, при якій між компонентами утворюються такі зв'язки, що забезпечують взаємодію і цілісність системи;
- варіативний склад обладнання, який передбачає наявність процесорного ядра і змінних апаратних модулів, за допомогою яких ядро адаптується до вимог конкретного використання;
- модульність, що передбачає розробку компонентів ІСУР у вигляді функціонально завершених модулів, що мають стандартний інтерфейс;
- відкритість програмного забезпечення і максимальне використання стандартних драйверів і програмного забезпечення;

Спеціалізація та адаптація апаратних і програмних засобів до структури алгоритмів обробки та формування команд управління

2 РОЗРОБКА СТРУКТУРНОЇ СХЕМИ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ УПРАВЛІННЯ

Адаптивність інтелектуальних роботизованих систем в Industry 4.0 заслуговує похвали, дозволяючи безперебійне налаштування відповідно до конкретних потреб наших різноманітних виробничих завдань. Крім того, впровадження передової робототехніки в нашу промислову структуру оптимізувало робочі процеси, дозволяючи пришвидшити час виконання робіт і підвищити загальну продуктивність [26].

У розробці інтелектуальних робототехнічних платформ ключовим аспектом є організація системного обслуговування, яке охоплює не лише технічне обслуговування компонентів, а й підтримку програмного забезпечення, оновлення моделей машинного навчання та збереження цілісності даних. Особливістю таких систем є можливість віддаленої діагностики, використання інтелектуальних агентів для моніторингу працездатності та впровадження стратегій профілактичного сервісу. Це забезпечує зниження простоїв, економію ресурсів та підвищення надійності платформи в цілому [26].

Метою роботи буде розробка системи взаємодії інтелектуальних датчиків, машин та інших роботизованих систем, з метою створення високовзаємопов'язаного та автоматизованого виробничого середовища.

2.1 Розробка структури ІСУР

Реалізація ІСУР для МРЦ заснована на аналізі та інтерпретації інформації з навколишнього середовища, а також на постійному відстеженні власних координат. У ІСУР буде використовуватися як пасивна, так і активна навігація. Пасивна навігація використовує інформацію про власні координати та інші характеристики руху МРП, яка надходить із зовнішніх джерел, а при активній навігації власні координати визначаються виключно за допомогою

приладів, розміщених на МРП.

ІСУР повинна забезпечувати такі види контролю:

- дистанційне керування;
- інтелектуальний гібрид;
- інтелектуально автономний.

Дистанційне керування МРП призначене для прийому, аналізу команд дистанційного керування від оператора та формування керуючих сигналів для системи руху. Особливість дистанційного керування МРЦ полягає в:

- необхідність підтримки постійного каналу зв'язку;
- обмеження радіуса управління рухом МРП (тільки в межах радіуса каналу зв'язку);
- неможливість самостійного виготовлення;
- затримки в управлінні командами управління;
- здійснення помилкових дій при втомі оператора.

Інтелектуальне автономне управління рухом МРП передбачає моделювання навколишнього середовища з різним ступенем деталізації, локалізацію місця розташування МРП, контроль руху, виявлення та уникнення перешкод. Особливість такого управління полягає в його використанні поза каналом зв'язку.

Для ефективного управління переміщенням МРП доцільно використовувати інтелектуальне гібридне управління, яке поєднує в собі переваги як віддаленого, так і інтелектуального автономного управління. Недоліком такого управління трафіком є те, що він працює тільки в радіусі каналу зв'язку.

Структура ІСУР показана на рисунку 2.1.

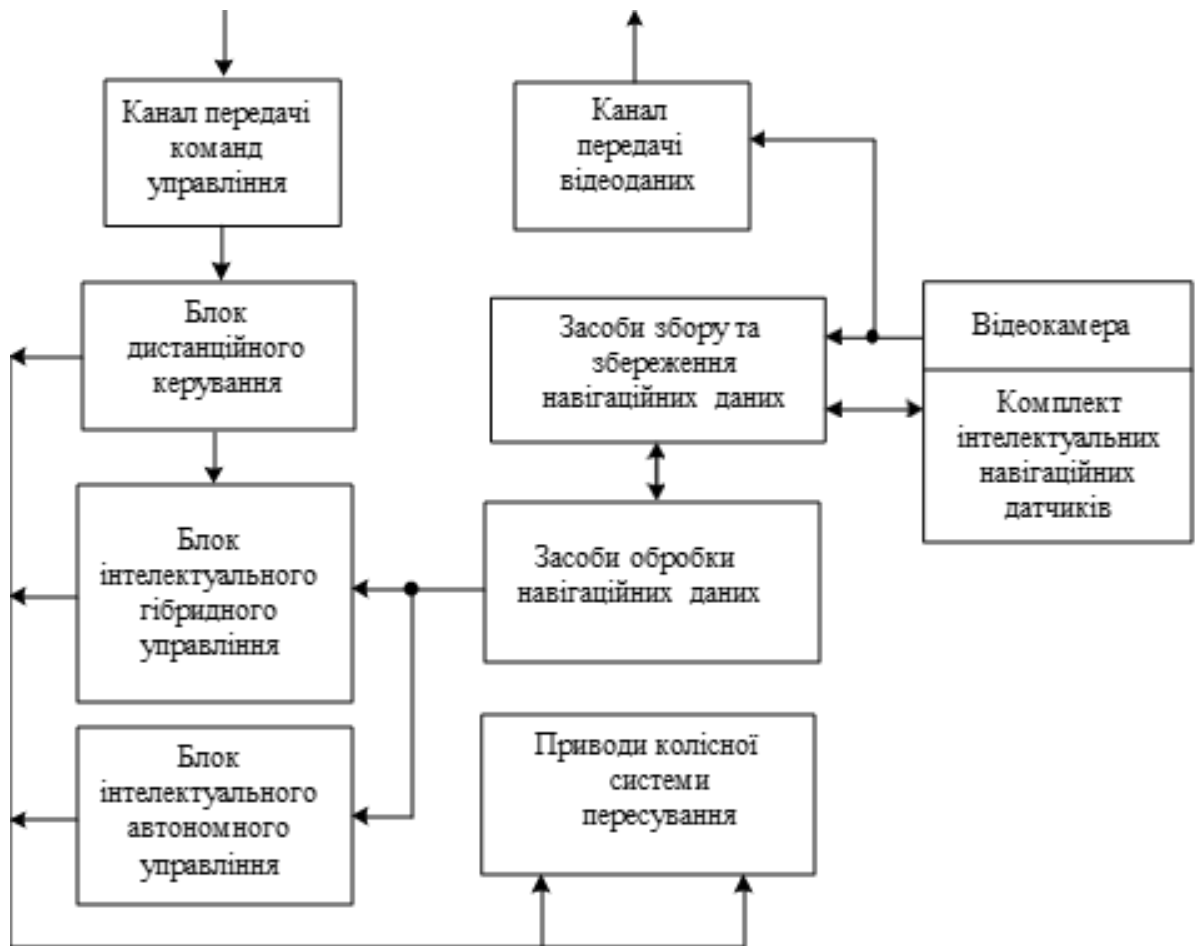


Рисунок 2.1 – Структура ІСУР

Основними компонентами ІСУР є: інтелектуальні автономні та інтелектуальні гібридні блоки управління, засоби збору, збереження та обробки навігаційних даних, відеокамера, комплект інтелектуальних навігаційних датчиків, канали передачі відео та передачі команд управління, блок дистанційного керування.

Однією з важливих функцій ІСУР є побудова моделі середовища, в якому рухається платформа. Дана модель навколишнього середовища повинна адекватно відображати поточний стан і положення платформи, можливі перешкоди для руху і небезпечні ділянки шляху. При створенні моделі навколишнього середовища використовуються різноманітні пристрої з різноманітними технологіями – ультразвукові, оптичні і т.д., 2D і 3D лідари. Електронні акселерометри, компаси та системи позиціонування на основі GPS використовуються для просторової орієнтації та кріплення MRP на місцевості.

Електронні акселерометри, компаси та системи позиціонування на основі технології GPS використовуються для просторової орієнтації та закріплення MRP на місцевості. MRP при внутрішньому розташуванні майже повністю покладаються на дані лідарів, акселерометрів і компасів. Обробка даних з датчиків і побудова моделі зовнішнього середовища реалізується модулем мікрокомп'ютера на базі SOC.

Запропоновано використовувати нечіткі методи та дистанційні інтелектуальні датчики на основі технології лазерної ехолокації для керування рухом колісного МРП. Існуючі лазерні датчики забезпечують роботу як при низькому, так і при високому рівні зовнішнього освітлення. Такі датчики забезпечують високу точність вимірювання відстані до чорних об'єктів на відстані до 30 метрів і до світлих об'єктів на відстані до 65 метрів.

Канал передачі команд управління в MRP побудований за допомогою вдосконалених промислових модулів: керуючого мікроконтролера, призначеного для узгодження інтерфейсу SPI трансивера з послідовним інтерфейсом керуючого мікрокомп'ютера. Керуючий мікроконтролер забезпечує координацію швидкості прийому/передачі даних шляхом їх буферизації.

ІСУР може управлятися за допомогою 8, 12, 16, 20, 24 і більше бітових команд. Для реалізації управління MRP розроблена 16-бітна команда. У команді управління перші вісім цифр (1-8) коду операції (ноги, рух вперед, назад, рух вліво, рух вправо), а другі вісім цифр (9-16) – параметри операції (швидкість, ступені обертання, відстань і час переміщення).

2.2 Алгоритми нечіткої логіки та засоби керування МРП

Нечітка логіка (НЛ) використовується в ситуаціях, де існує високий ступінь невизначеності, складності і нелінійності. Нечіткий контроль означає використання НЛ для класифікації даних і прийняття управлінських рішень, що забезпечують уникнення таких проблем з навігацією, як безперервні петлі,

зворотне відстеження, тупикові пастки і рух у вузьких проходах.

Будова засобів метушливого управління рухом коліс МРП наведена на рисунку 2.2.



Рисунок 2.2 – Структура засобів нечіткого логічного управління рухом МРП

Основними складовими засобами нечіткого управління рухом МРП є датчики і програмне забезпечення НЛ-контролера, який складається з блоків фазифікації, прийняття рішень, дефазифікації і зводу правил.

Зміна швидкості і траєкторії руху МРП здійснюється на основі інформації, що надходить від бортових інтелектуальних навігаційних датчиків. Дані з бортових інтелектуальних навігаційних датчиків надходять на контролер програми FL, в якому вони перетворюються в керуючі сигнали, за допомогою яких здійснюється управління рухом МРЦ. Програмний контролер НЛ працює з нечіткими величинами, до яких відносяться нечіткі множини, нечіткі змінні і лінгвістичні змінні.

Нечітка множина A в деякому просторі X задається множиною пар:

$$A = \{(x, \mu_A(x)); x \in X\}, \quad (3.1)$$

де $\mu_A(x)$ це функція приналежності, яка показує ступінь належності кожного елемента до нечіткої множини A .

Максимальне значення функції належності дорівнює 1, що означає повну належність елемента до нечіткої множини.

Нечітка змінна має множину $\langle \alpha, X, A \rangle$, де α — ім'я нечіткої змінної; X — область його визначення; A — нечітка множина, яка містить можливі значення, які можуть приймати нечітку змінну α .

Мовна змінна задається множиною $\langle \beta, T, X, G, M \rangle$, де β — ім'я мовної змінної; T — сукупність значень мовної змінної (терм-множина); X — область визначення нечітких змінних; G — синтаксична процедура, що дозволяє генерувати нові терміни (значення) за допомогою елементів Т-сеттинга; M — семантична процедура, яка дозволяє новому значенню лінгвістичної змінної, отриманої за допомогою процедури G , відповідати нечіткій множині.

Основними етапами функціонування програмного контролера НЛ є: фазифікація, прийняття управлінських рішень і дефазифікація. Розглянемо кожен етап роботи програмного контролера НЛ більш детально.

На етапі фазифікації вхідні дані $x = (x_1, x_2, \dots, x_n)$, що надходять від інтелектуальних навігаційних датчиків, перетворюються з чітких значень в нечіткі. Таке перетворення здійснюється шляхом порівняння питомого значення вхідного значення і значення функції $\mu_A(x)$ відповідного члена вхідної лінгвістичної змінної. Особливість етапу фазифікації полягає у формуванні функцій приналежності та визначенні кількості членів мовної змінної.

Вхідна лінгвістична змінна — «відстань до перешкоди», яка формується на основі даних від інтелектуальних навігаційних датчиків, може бути використана для управління рухом МРП.

На виході блоку фазифікації можуть бути сформовані значення лінгвістичної змінної (терміна), наприклад, $T = \{\text{"маленький"}, \text{"середній"}, \text{"великий"}\}$.

Етап прийняття управлінських рішень ґрунтується на нечітких правилах. Особливості реалізації даного етапу полягають у забезпеченні достатніх обсягів і узгодженості нечітких правил, які повинні адекватно

відображати мету і охоплювати всі можливі випадки управління рухом МРЦ. Для програмного забезпечення НЛ правила записуються в наступному вигляді:

$$P_k(B): (X_l - \text{це } A_{lk} \text{ і } \dots X_n \in A_{nk}) \text{ або } (X_l \in A_{lm} \text{ і } \dots \text{ а } X_n \in A_{nm}), \quad (3.2)$$

де X_i – вхідні змінні, R_k – правило, A_{ik} , A_{im} , B_k – термін лінгвістичні змінні.

При побудові нечітких правил спочатку необхідно розрізняти певні терміни в кожній вихідній лінгвістичній змінній і встановлювати їх функції. Наступним кроком для кожного члена вихідної лінгвістичної змінної є правило з використанням кон'юнкції (операція "і") і диз'юнкції (операція "або"). Так, для вихідної лінгвістичної змінної значення "кут повороту", тобто її термами можуть бути: $T = \{ "right", "forward", "left" \}$. Наприклад, для терміна "правильно" буде записано нечітке правило:

Правило 1 («Праворуч»): (якщо «відстань до перешкоди зліва» — «мала», а «відстань до перешкоди спереду» — «велика», а «відстань до перешкоди справа» — «велика») або («відстань до перешкоди зліва» є « середня» і « Відстань до перешкоди справа » є « середнім »).

На етапі прийняття рішень виконується перетворення вхідної нечіткої множини в вихідну нечітку множину з використанням нечіткої бази правил відповідно до наступного виразу:

$$R_k = M_k \rightarrow N_k \quad (3.3)$$

де правило R_k – це правило, M_k , N_k – вхідні та вихідні нечіткі множини.

На стадії дефазифікації перетворення нечіткого значення N_k вихід, що надходить від блоку прийняття рішень, проводиться в чітке значення, яке подається на привід колісної системи руху МРЦ, відповідно до наступного виразу:

$$y_k = F(N_k), \quad (3.4)$$

де F – алгоритм розфазифікації.

3 РЕАЛІЗАЦІЯ СТРУКТУРИ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ УПРАВЛІННЯ КОЛІСНИМ МРП

Реалізація структури інтелектуальної системи управління колісним МРП здійснюється за допомогою окремих функціональних блоків. Для зведення зазначених блоків використовувалися готові модулі. На сьогоднішній день номенклатура модулів, наприклад, для побудови приводів колісної системи руху МРП, досить широка і забезпечує адекватне управління двигунами відносно великої потужності.

При розробці системи управління в дистанційному режимі здійснюється програмна розробка блоків дистанційного керування, інтелектуального гібридного керування, інтелектуального автономного керування. Для цього використовується ноутбук, на якому запускається і налагоджується нечітка система управління. В онлайн-режимі актуальна інформація з датчиків передається на ноутбук через мережу Wi-Fi.

Ця інформація обробляється розроблюваною системою управління, яка передає керуючі сигнали по тій же мережі Wi-Fi на колісну МРП. Такий підхід дозволяє швидко розробити і налагодити систему управління без необхідності постійно завантажувати оновлене програмне забезпечення на мобільну платформу.

Надалі розроблені і випробувані програмні засоби з урахуванням кросплатформенного характеру використовуваних засобів розробки можуть бути легко завантажені в керуючий мікрокомп'ютер для регулярного використання.

Інтерфейс з використанням керуючих сигналів GPIO використовується для подачі команд на приводи колісної системи пересування. Блок управління реалізований на базі мікроконтролера типу ESP-32, основним завданням якого є прийом і виконання команд щодо напрямку руху, визначення швидкості руху і повороту колісної системи МРП.

Виявлення перешкод по ходу руху здійснюється за допомогою індивідуальних датчиків, які охоплюють поле зору перед платформою і призначені для реагування на такі перешкоди. Така система побудована на базі двох безконтактних датчиків відстані, які забезпечують виявлення перешкод на відстані до 1-2 метрів.

Канал передачі команд управління для тестування MRP реалізований за допомогою мережі Wi-Fi. В результаті використання MQTT сервера (брокера) забезпечується можливість віддаленого збору даних з усіх датчиків MRP, що забезпечує можливість розробки, побудови та налагодження моделі зовнішнього середовища та програмного забезпечення блоків дистанційного керування, інтелектуального гібридного керування та інтелектуального автономного керування у віддаленому режимі на основі поточної інформації від датчиків.

Надалі розроблені і випробувані програмні засоби завантажуються на керуючий мікрокомп'ютер для регулярного використання з урахуванням кросплатформенного характеру використовуваних засобів розробки.

Для управління рухом доцільно використовувати готові промислові модулі на базі сімейств мікроконтролерів ESP8266 і ESP32. Основними перевагами цих сімейств мікроконтролерів є низька вартість, доступність засобів розробки програмного забезпечення (Espressif IoT Development Framework), PlatformIO, LUA і т.д.), наявність плат розширення для прийому даних з датчиків і управління різними пристроями і т.д.

На мікроконтролері ESP32 реалізовані спеціалізовані модулі, які є основою для побудови готових апаратних рішень. На рис. 3.1 показаний загальний вигляд модуля виявлення перешкод на базі ESP32 C3, виконаного за концепцією IoT.

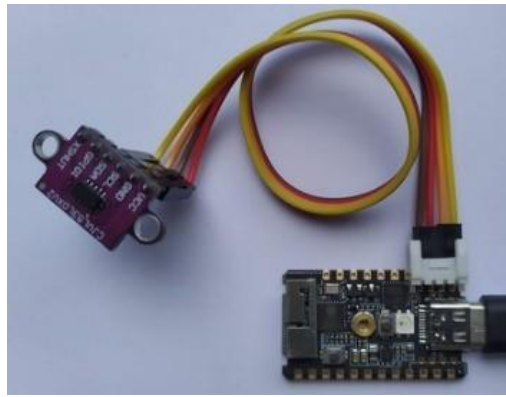


Рисунок 3.1 – Вигляд модуля датчика віддаленого від перешкоди VL53L0X на базі плати M5Stamp з мікроконтролером ESP32-C3

На рисунку 3.2 наведено графік даних з датчика відстані VL53L0X при його розміщенні на відстані 100 мм від перешкоди та гістограма розподілу змін цих даних (рисунок 3.3).

Модулі тестувалися за допомогою сайту thingspeak.com. На сайті передбачено зберігання даних мікроконтролерних систем та їх відображення у вигляді графіків, діаграм, гістограм після аналізу в середовищі MatLab.

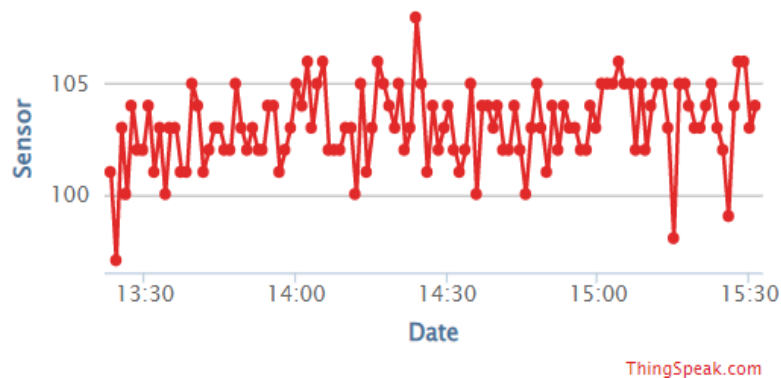


Рисунок 3.2 – Отримані дані про відстань від датчика VL53L0X

Вимірювання відстані за допомогою датчика відстані VL53L0X відбувається за результатами аналізу гістограми розподілу з точністю, прийнятною для роботи МРЦ. При фактичній відстані в 100 мм розкид становить не більше 3 мм, як впливає з аналізу гістограми, існує похибка присадки в 2 мм, яку можна усунути шляхом корекції показань і калібрування приладу.

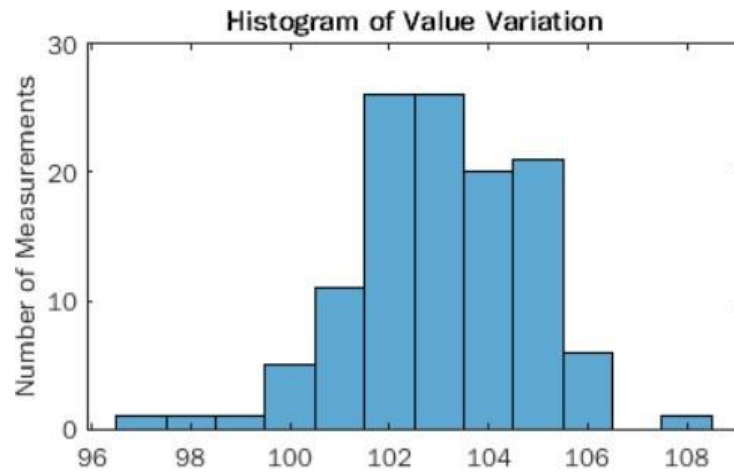


Рисунок 3.3 – Гістограма розподілу значень від датчика відстані VL53L0X

Також було перевірено час передачі даних на віддалений thingspeak.com сервер за протоколом MQTT. При цьому оцінювався час від моменту формування команди «опублікувати» до отримання даних згідно з підпискою. Гістограма розподілу в часі наведена на рисунку 3.4.

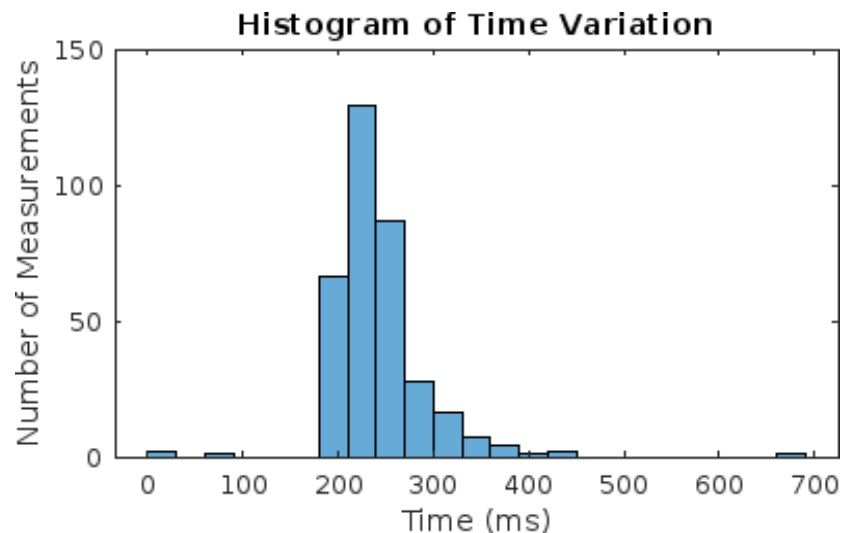


Рисунок 3.4 – Гістограма розподілу часу відгуку для команди MQTT "publish" для віддаленого сервера mqtt3.thingspeak.com

Показання вбудованого годинника мікроконтролера використовувалися для вимірювання значень заданих параметрів. Далі була побудована гістограма розсіювання параметрів за допомогою інструментів сайту Matlab.

Проведений аналіз дозволяє констатувати, що зазначений час реакції в середньому становить 250 мс. Результат показаний на рис. 3.5.

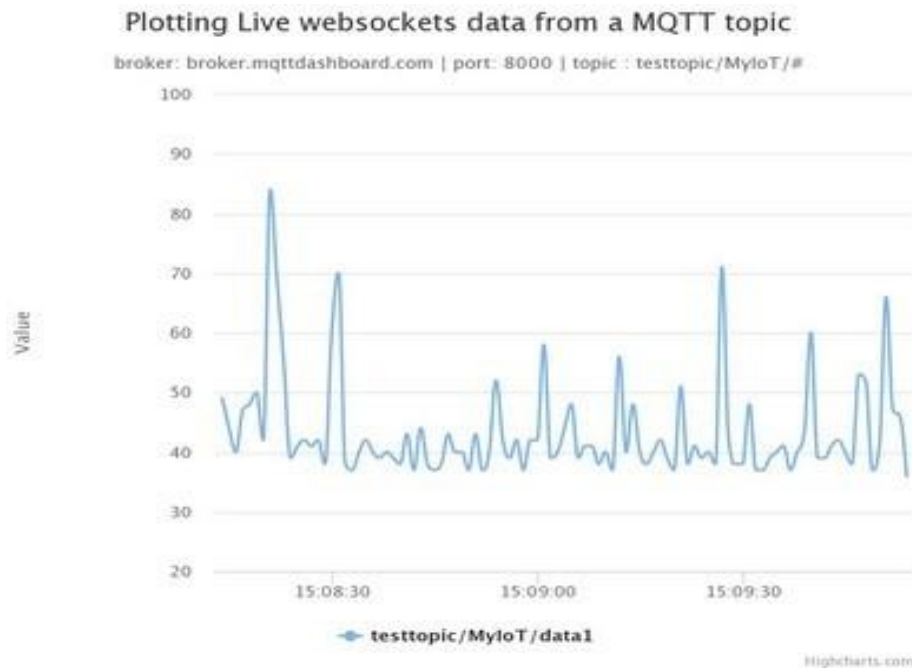


Рисунок 3.5 – Час відгуку команди "publish" для сервера broker.mqtt-dashboard.com

Слід зазначити, що задані значення виходять для фактичного віддаленого сервера mqtt3.thingspeak.com, фізично розташованого на іншому континенті. Для тестового сервера broker.mqtt-dashboard.com аналогічний параметр становив близько 45 мс.

Тому зазначені значення затримки передачі даних від датчиків близько 50 мс слід вважати прийнятними для реалізації процесу віддаленого налагодження програмного забезпечення. Ця затримка може бути значно зменшена при використанні локального сервера і протоколу UDP.

Моделювання керування нечіткою логікою для ІСУР було проведено з використанням середовища JuzzyOnline. Це браузерний інструментарій для проектування та виконання систем нечіткої логіки типу 1, інтервального та загального типу 2. JuzzyOnline включає в себе функції генерації фігур для всіх типів нечітких множин [27].

Онлайн-версія на основі Juzzy – інструментарію java для розробки систем нечіткої логіки [28]. Системний вигляд нечіткої системи керування показаний на рисунку 3.6.

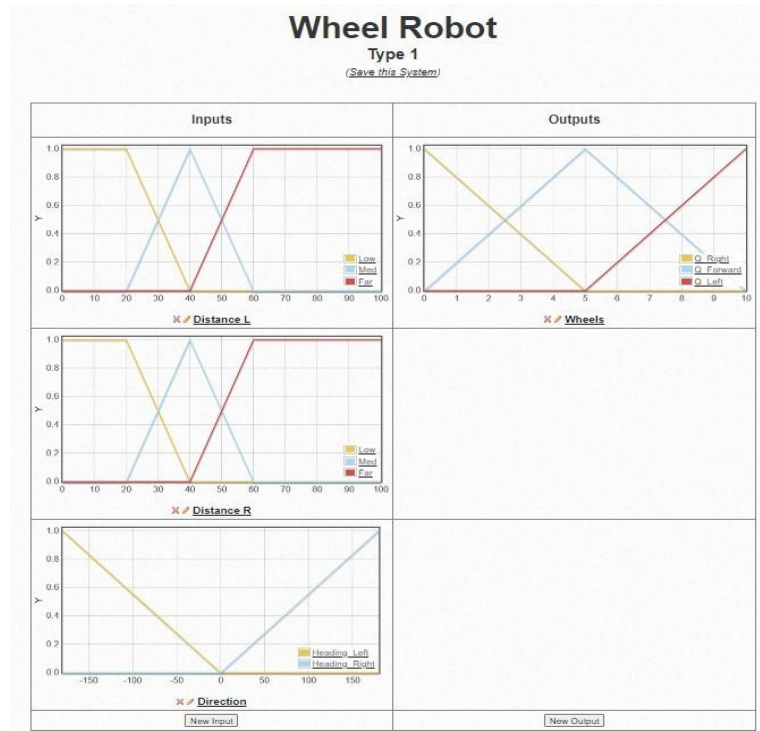


Рисунок 3.6 – Системний вигляд у середовищі JuzzyOnline

Використовуються системи нечіткої логіки типу 1. У системах нечіткої логіки типу 1 механізм логічного висновку об'єднує правила та надає відображення від вхідних нечітких множин типу 1 до вихідних нечітких множин типу 1. Кілька антецедентів у правилах з'єднуються відповідно до перетину множин. Оцінки членства у вхідних наборах поєднуються з оцінками у вихідних наборах за допомогою композиції sup-star.

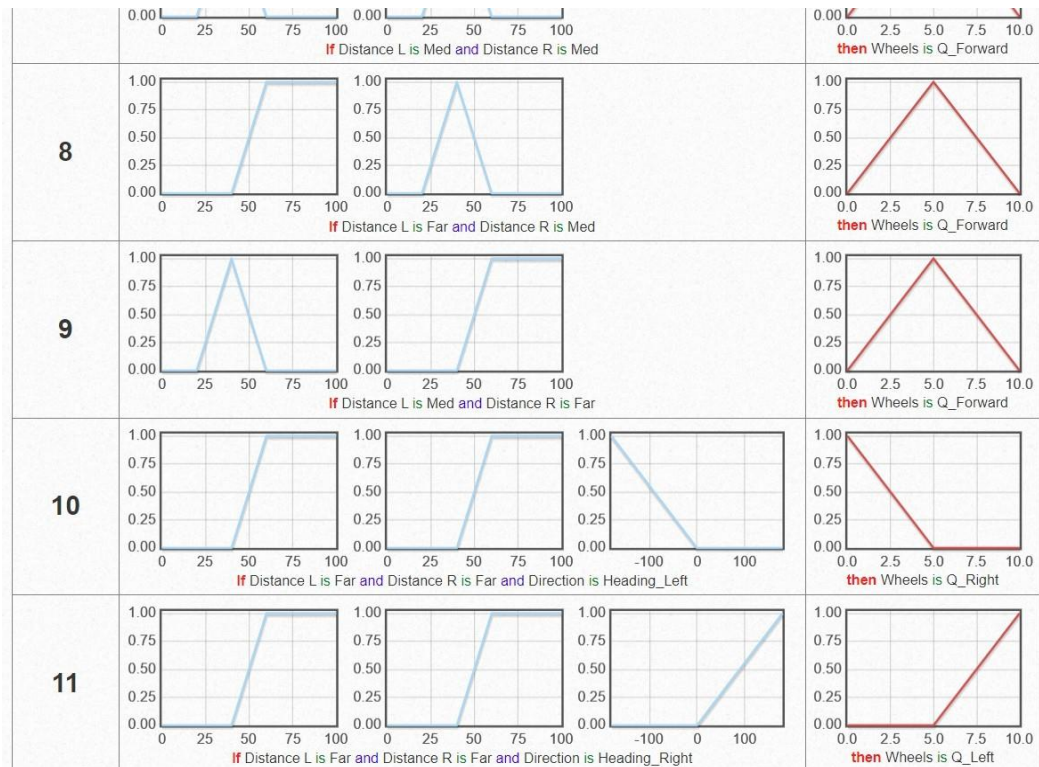
Множинні правила можуть бути об'єднані за допомогою операції, що відповідає об'єднанню множин, або при дефазифікації методом зваженого підсумовування.

Моделювання було виконано для випадку MRP, який має два датчики відстані, розташовані спереду, праворуч і ліворуч від центральної осі MRP. Вони використовуються для виявлення перешкод у напрямку руху платформи, які розташовані на відстані до 1 м. Для установки загального напрямку руху

використовується електронний датчик гіроскопа.

У цьому випадку під час моделювання дані з датчика обробляються таким чином, щоб бажаний напрямок руху відповідало 0 градусам, відповідно, відхилення можуть становити 180 градусів в обидві сторони.

Частина набору антецедентів і послідовних правил Mamdani, реалізованих в середовищі JuzzyOnline, показана на рисунку 3.7.



Малюнок 3.7 – Правило Мамдані антецедент і наслідок

Вхідними параметрами є відстань до перешкоди, отримана від виносних датчиків, і напрямок руху. Для опису вхідних параметрів використовуються лінгвістичні змінні Distance L і Distance R, які відповідають відстані до перешкоди, отриманої від двох датчиків відстані.

Лінгвістична змінна Direction описує параметр напрямку руху в діапазоні від -180 до 180 градусів.

Далі, в середовищі JuzzyOnline, для кожної вхідної змінної необхідно встановити набір значень (термів) в позначенні $T=\{ "Low", "Med", "High" \}$, що позначають малу, середню та велику відстань до перешкоди відповідно.

Позначення $T=\{"Heading_Left", "Heading_Right"\}$ вказує напрямок і величину відхилення MRTP від обраного курсу. Далі слід охарактеризувати

Функції приналежності кожного члена задаються парами чисел, перше з яких вказує на величину відстані до перешкоди. Результатом обробки є кут повороту MRP, який описується лінгвістичною змінною *Wheels* з трьома термінами $T=\{"Q_Right", "Q_Forward" \text{ та } "Q_Left"\}$, які означають правий поворот, прямий рух та лівий поворот.

При цьому початкове значення Коліс, рівне 0, відповідає відсутності руху, а 1 - руху з максимальною швидкістю. Вхідні значення в діапазоні від 0 до 5,0 інтерпретуються як приналежність до управління лівого двигуна МРП, а в діапазоні від 5,0 до 10,0 – правого. Функції членства даються для кожного з термінів. Показані відповідні функції приналежності вхідних лінгвістичних змінних та виводу.

Далі необхідно було виконати формування бази правил за допомогою середовища JuzzyOnline. Формування бази правил здійснюється для кожного члена вихідної лінгвістичної змінної з використанням вхідних лінгвістичних змінних, їх термінів і відповідних операцій.

Сформована база правил показана на рисунку 3.8.

Rules
✗ 1. If Distance L is Far and Distance R is Far then Wheels is Q_Forward
✗ 2. If Distance L is Low and Distance R is Low then Wheels is Q_Right
✗ 3. If Distance L is Low and Distance R is Med then Wheels is Q_Right
✗ 4. If Distance L is Low and Distance R is Far then Wheels is Q_Right
✗ 5. If Distance L is Med and Distance R is Low then Wheels is Q_Left
✗ 6. If Distance L is Far and Distance R is Low then Wheels is Q_Left
✗ 7. If Distance L is Med and Distance R is Med then Wheels is Q_Forward
✗ 8. If Distance L is Far and Distance R is Med then Wheels is Q_Forward
✗ 9. If Distance L is Med and Distance R is Far then Wheels is Q_Forward
✗ 10. If Distance L is Far and Distance R is Far and Direction is Heading_Left then Wheels is Q_Right
✗ 11. If Distance L is Far and Distance R is Far and Direction is Heading_Right then Wheels is Q_Left
<input type="button" value="New Rule"/>

Рисунок 3.8 – База правил в середовищі JuzzyOnline

Результати моделювання руху колісного МРТ з різними значеннями показань датчиків у навколишньому середовищі на основі розроблених правил

наведені на рисунку 3.9.

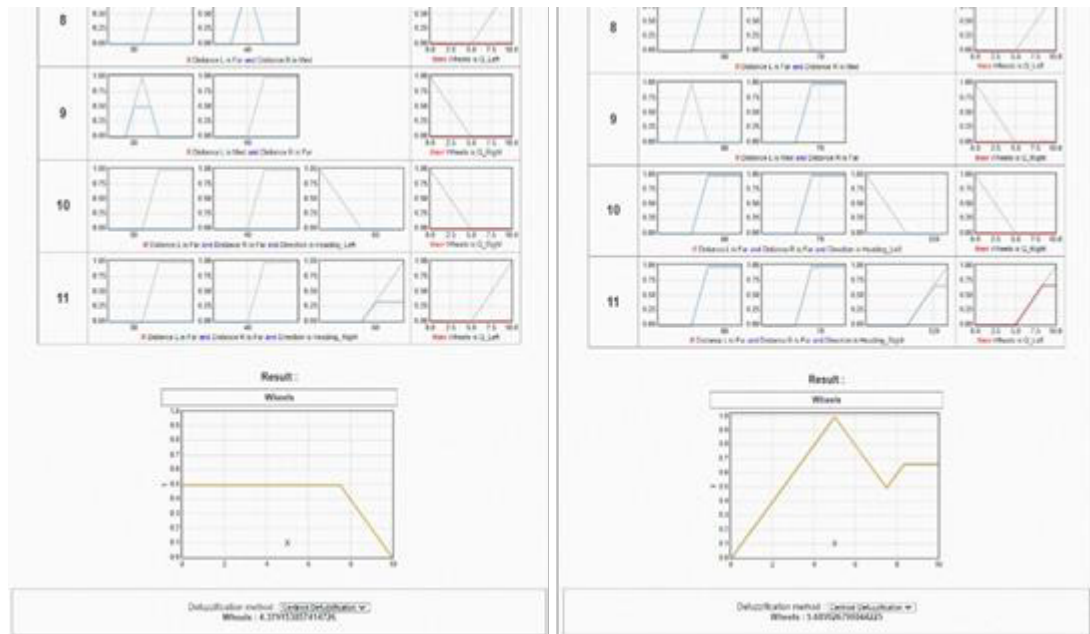


Рисунок 3.9 – Результати моделювання нечіткої системи керування

Розроблені результати моделювання нечіткої системи керування в режимі Surface View представлені на рисунку 4.10. Поверхня була побудована при значеннях показань датчиків з назвами Distance R, Distance L і при значенні 100 градусів від датчика з іменем Direction.

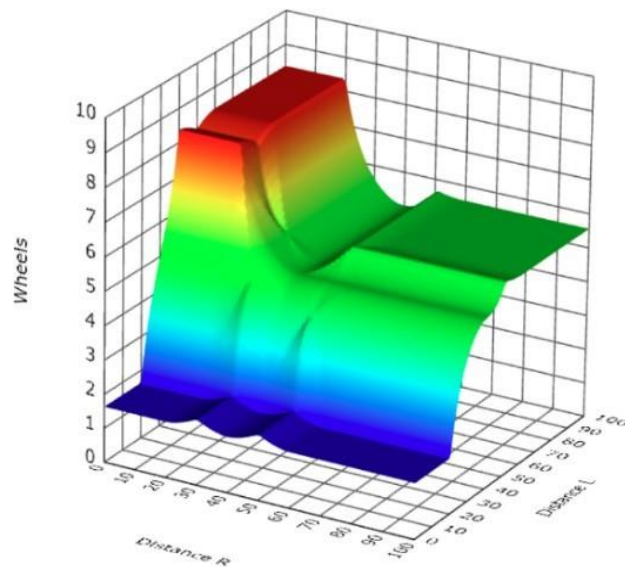


Рисунок 3.10 – Модель системи нечіткого керування

Моделювання системи нечіткого керування призводить до виконання в

режимі Surface View для параметрів відстані R, відстані L при напрямку 100 градусів. Як видно, для заданих значень відбувається зміна вихідного значення колісного контролера, що визначає різну швидкість обертання пари коліс МРП і, відповідно, зміна її руху як реакцію на наявність перешкод або відхилення від обраного напрямку руху.

4 РОЗРОБКА РОБОТИЗОВАНОЇ ПЛАТФОРМИ

4.1 Визначення архітектури

У секторі апаратного забезпечення з'явилися високошвидкісні, недорогі ПК, швидкі відеоплати з 3D-графікою та недорогих карт керування рухом. Отже, платформа ПК тепер забезпечує різнобічний функціонал, а отже, робить складні програмні архітектури та фірмові апаратні компоненти зайвими в більшості випадків. QMotor Robotic Toolkit (QMotor RTK), наприклад, інтегрується в режимі реального часу керування маніпулятором і графічний інтерфейс користувача (GUI) на одній платформі ПК.

Незважаючи на широку функціональність платформи ПК, велика частина досліджень в області програмного забезпечення для управління роботами використовує розподілені архітектури. Крім очевидних переваг розподілених систем (наприклад, більша обчислювальна потужність), є ряд недоліків. Зокрема, розподілена архітектура значно підвищує складність програмного забезпечення. Крім того, жорстка поведінка в реальному часі через мережеві з'єднання часто вимагає дорогого пропрієтарного обладнання. Як правило, загальна вартість обладнання вища, і користувачам доводиться ознайомлюватися з різними апаратними архітектурами та операційними системами. Незважаючи на те, що багато платформ, розроблених за останні кілька років, намагалися бути розширюваними, ці платформи рідко використовуються та використовуються повторно.

З огляду на вищезазначені зауваження, платформа, яка була розроблена для інтеграції контурів управління сервоприводами, генерації траєкторій, програм на рівні завдань, програм з графічним інтерфейсом і 3D-моделювання в однорідну програмну архітектуру. Тобто використовується тільки одна апаратна платформа (ПК), одна операційна система (QNX Real-Time Platform) і одна мова програмування (C++).

Даний тип архітектури має наступні переваги:

– простота: однорідна нерозподілена архітектура набагато менша і простіша, ніж розподілена неоднорідна архітектура. Він простіший у встановленні, простіший для розуміння та простіший у розширенні;

– гнучкість на всіх рівнях: кожен компонент роботизованої платформи відкритий для розширень і модифікацій. Багато минулих платформ використовували відкриту архітектуру на деяких рівнях, але інші рівні (наприклад, сервокерування) були реалізовані на власному обладнанні таким чином, що їх не можна було модифікувати;

– вартість: роботизована платформа вимагає менше апаратних компонентів, ніж розподілена платформа. По суті, достатньо одного ПК з однією або кількома платами вводу/виводу (I/O). Крім того, апаратне забезпечення ПК є дуже економічно вигідним.

4.2 Визначення інструментів розробки

Щоб зменшити зусилля та складність розробки, роботизована платформа базується на інструментах та технологіях загального призначення.

4.2.1 Робоча станція

У той час як в минулому тільки дорогі робочі станції UNIX забезпечували обчислювальну потужність, необхідну для управління роботизованими системами, ПК наздогнали або навіть перевищили продуктивність робочих станцій. У порівнянні з робочими станціями UNIX, система на базі ПК дозволяє використовувати більшу різноманітність апаратних і програмних компонентів. Крім того, ці компоненти та сам ПК зазвичай дешевші, ніж їх аналоги UNIX.

4.2.2 Платформа реального часу QNX

Платформа реального часу QNX (RTP) від QSSL складається з операційної системи QNX6/Neutrino та додаткових компонентів для розробки

та мультимедіа. QNX6 — це вдосконалена операційна система реального часу, яка забезпечує сучасну архітектуру на основі мікроядер, POSIX-сумісний програмний інтерфейс, самостійну розробку, можливості 3D-графіки та просту архітектуру драйвера пристрою. RTP також дуже економічно вигідний, оскільки він безкоштовний для некомерційного використання та працює на недорогих стандартних ПК.

4.2.3 Об'єктно-орієнтоване програмування на C++

Що стосується розробки програмного забезпечення для управління роботами, то об'єктно-орієнтоване програмування має ряд переваг перед процедурним програмуванням. По-перше, він надає мовні конструкції, які дозволяють набагато простіше програмувати інтерфейс. Наприклад, множення матриць може бути виражене простим «*», подібно до програмування MATLAB. По-друге, об'єктно-орієнтоване програмування дозволяє створити архітектуру системи, яка є дуже гнучкою, але в той же час простою. Тобто компоненти (класи) системи можуть мати вбудовану поведінку за замовчуванням і налаштування за замовчуванням. Програміст може використовувати цю поведінку за замовчуванням для зменшення розміру коду або його перевизначення для конкретних програм. Нарешті, об'єктно-орієнтоване програмування підтримує узагальнене програмування, що полегшує розробку компонентів, незалежних від конкретної реалізації (наприклад, загальний клас "Маніпулятор" буде працювати з різними типами маніпуляторів).

Всі перераховані вище переваги засновані на загальних поняттях об'єктно-орієнтованого програмування: абстракція; інкапсуляція; поліморфізм; успадкування.

Мовою вибору є C++, оскільки вона надає весь спектр об'єктно-орієнтованих понять при збереженні високої продуктивності.

4.2.4 Анімація об'єктів

Open Inventor, розроблений компанією Silicon Graphics, є об'єктно-орієнтованою бібліотекою C++ для створення та анімації 3D-графіки. Open Inventor мінімізує зусилля на розробку, оскільки здатний завантажувати 3D-моделі, які створюються у форматі Virtual Reality Modeling Language (VRML). Доступні різноманітні програмні пакети, які полегшують побудову 3D VRML-моделей, що представляють роботизовані компоненти. Роботизована платформа також використовує функціональність Open Inventor для анімації цих компонентів.

4.2.5 Використання системи QMotor

Реалізація стратегій управління вимагає здатності встановлювати детермінований контур управління в режимі реального часу, реєструвати дані, налаштовувати параметри управління і будувати графік сигналів. Для цього використовується графічне керуюче середовище QMotor для роботизованої платформи.

4.3 Компоненти роботизованої платформи

Кожен компонент роботизованої платформи (маніпулятори, генератор траєкторій тощо) моделюється класом C++. Визначення класу C++ поєднує дані та функції, пов'язані з цим компонентом. Наприклад, клас "Puma560" містить дані робота Puma 560 (поточне положення суглоба), а також функції, пов'язані з Puma (включення живлення руки). Таким чином, дизайн роботизованої платформи є результатом групування даних і функцій у кілька класів осмисленим та інтуїтивним способом. Клас може використовувати частини функціональності та дані іншого класу (так званого базового класу), виводячи цей клас з базового класу. Цей процес називається успадкуванням, і він значною мірою пов'язаний з повторним використанням коду та усуває надмірність у системі. Для розширення системи програміст створює нові класи. Зазвичай, нові класи будуть отримані з одного з уже існуючих класів,

щоб мінімізувати зусилля з кодування.

Класи роботизованої платформи включають компоненти графічного інтерфейсу та 3D-моделювання для графічного моделювання. Ці типи компонентів традиційно зустрічалися в окремих програмах (наприклад, див. симулятор робота RCCL). Однак, включивши їх в один клас, ми можемо досягти тісної інтеграції графічного інтерфейсу, 3D-моделювання та інших функціональних частин. Крім того, системні розширення автоматично включають компоненти графічного інтерфейсу та 3D-моделювання.

Діаграма ієрархії основних класів роботизованої платформи показана на рисунку 4.1.



Рисунок 4.1 – Класова ієрархія роботизованої платформи

Кожна стрілка малюється від похідного класу до батьківського класу; Отже, чим лівіше вказано клас, тим він є більш загальним. Класи роботизованої платформи можна розділити на такі категорії:

Основні класи. Класи `RoboticObject`, `FunctionalObject` та `PhysicalObject` складають основу всіх роботизованих об'єктів. Класи `RoboticPlatform` і

ObjectManager містять функціонал для загального управління програмами управління роботами.

Загальні класи робототехніки. Похідними від основних класів є ряд загальних класів роботів. Ці класи не можуть бути реалізовані. Скоріше, ці класи служать базовими класами, які реалізують загальну функціональність, а також представляють загальний інтерфейс програмісту (тобто, ці класи можуть використовуватися для створення програм, які не залежать від конкретного апаратного забезпечення або конкретного алгоритму).

Специфічні класи робототехніки. Похідними від загальних класів роботів є класи, які реалізують певну частину апаратного забезпечення (наприклад, клас Puma560 реалізує робота Puma 560) або конкретну функціональну складову (наприклад, клас DefaultPositionControl реалізує контроль положення пропорційної інтегральної похідної (PID)).

Клас ControlProgram. Цей клас є частиною системи QMotor. Він є основою для всіх контурів управління в реальному часі. Класи, які потребують циклу керування в реальному часі, є похідними від класу ControlProgram.

Крім класів, показаних на рисунку 4.1, роботизована платформа надає класи математичної бібліотеки, класи моделей маніпулятора та кілька утилітарних класів. Ці класи та їх класова ієрархія будуть описані далі в цій статті.

У програмі управління роботом програміст створює екземпляри об'єктів з класів. Програміст може створити екземпляр стільки об'єктів, скільки забажається з одного класу. Наприклад, керувати двома роботами Puma просто, просто створивши два об'єкти класу Puma560. Як тільки об'єкти створені, програміст може використовувати їх функціонал. Менеджер об'єктів (рисунок 4.2) веде список всіх поточних екземплярів об'єктів.

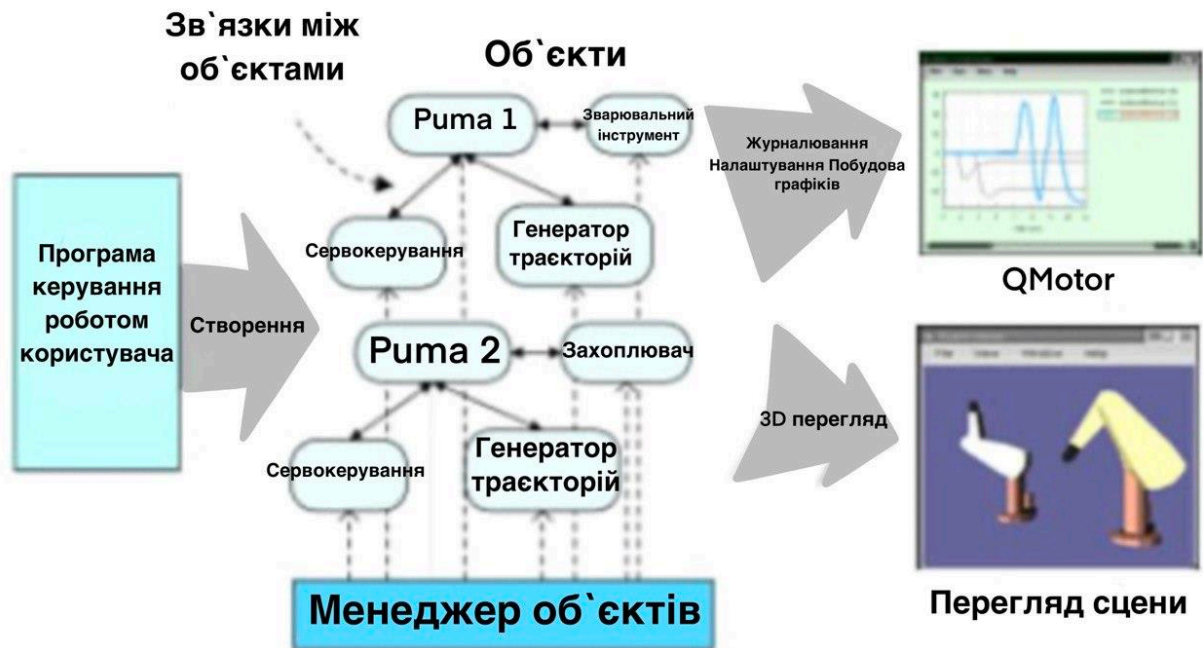


Рисунок 4.2 – Архітектура часу виконання роботизованої платформи

За допомогою менеджера об'єктів можна ініціювати функціональність на кількох об'єктах (наприклад, вимкнути всі об'єкти). Scene Viewer є графічним інтерфейсом за замовчуванням роботизованої платформи. Він містить вікна для перегляду 3D-сцени роботизованої робочої камери та список усіх об'єктів. Графічний інтерфейс QMotor може бути додатково використаний для реєстрації даних, побудови графіків та налаштування параметрів керування.

У роботизованій системі різні компоненти пов'язані один з одним. Для відображення цього факту встановлюються об'єктні відносини між об'єктами. Наприклад, об'єкти можуть вказувати їх фізичний зв'язок один з одним. Зв'язки між об'єктами реалізуються за допомогою вказівників C++ на пов'язаний об'єкт. Відносини між об'єктами позначені стрілками на рисунку 4.2.

Роботизована платформа використовує глобальний файл конфігурації для визначення конфігурації системи. Для кожного об'єкта у файлі

конфігурації у квадратних дужках вказано ім'я об'єкту, ім'я класу об'єкта та налаштування об'єкта (рисунок 4.3).

```
[leader]
class Puma560
position 0 0 0

[follower]
class BarrettArm
simulationMode on

[gripper]
class BarrettHand
port /dev/ser1
```

Рисунок 4.3 – Приклад глобального файлу конфігурації

4.4 Основні класи

Клас `RoboticObject` є базовим класом для всіх класів роботів. Він визначає загальний інтерфейс (тобто набір функцій, які можуть бути використані з усіма класами роботів Роботизованої платформи). Наприклад, програма може використовувати функцію `createControlPanel()` для повідомлення об'єкту класу `Puma560` або класу `Gripper` для відображення відповідної панелі керування. Зокрема, клас `RoboticObject` визначає i) обробку помилок, ii) обробку імен об'єктів, iii) управління конфігурацією, iv) вимкнення об'єктів, v) панель керування та vi) управління потоками. Зауважимо, що фактичний функціонал зазвичай є реалізовано в похідному класі. Однак, клас `RoboticObject` також реалізує простий функціонал за замовчуванням. Клас `PhysicalObject` є похідним від класу `RoboticObject`. Це базовий клас для всіх класів, який представляють фізичні об'єкти (наприклад, маніпулятори, датчики, захвати, і т.д.). Зокрема, клас `PhysicalObject` визначає наступну загальну функціональність:

- 3D візуалізація: кожен фізичний об'єкт може надати свою 3D-модель Open Inventor. Переглядач сцен циклічно перебирає всі фізичні об'єкти, щоб створити всю 3D-сцену;

- зв'язки об'єктів: фізичний об'єкт може вказувати інший об'єкт як місце монтажу. Використовуючи це співвідношення об'єктів, Scene Viewer може

малювати об'єкти в потрібному місці (наприклад, захват, встановлений на кінцевому ефекторі маніпулятора);

- положення та орієнтація: програміст може задати абсолютне розташування об'єкта в робочій комірці (або місце монтування, якщо вказано підключення об'єкта);

- режим симуляції: кожен фізичний об'єкт може бути заблокований у режимі симуляції. Тобто об'єкт не виконує ніякого апаратного введення-виведення; Замість цього моделюється його поведінка.

Клас `FunctionalObject` на даний момент не містить жодної функціональності. Він додається лише як симетричний аналог до класу `PhysicalObject`. Функціональні роботизовані класи, такі як клас `TrajectoryGenerator`, є похідними від класу `FunctionalObject`.

4.5 Класи, пов'язані з маніпуляторами

Центральними компонентами будь-якої роботизованої робочої клітини є маніпулятори. Клас `Маніпулятор` – це узагальнений клас, який визначає загальну функціональність маніпуляторів з будь-якою кількістю суглобів. Похідним від класу `Manipulator` є клас `DefaultManipulator`, який містить реалізацію за замовчуванням для маніпуляторів з відкритою архітектурою. Маніпулятори з відкритою архітектурою забезпечують доступ до поточного положення суглоба та керування крутним моментом/силою маніпулятора і, отже, дозволяють використовувати користувацькі алгоритми керування сервоприводом. Похідними від класу `DefaultManipulator` є класи, які реалізують конкретні типи маніпуляторів. В даний час підтримуються два маніпулятора: робот `Puma 560` і маніпулятор `Barrett Whole Arm Manipulator (WAM)` як в 4-ланковій, так і в 7-ланковій конфігурації.

Клас `DefaultManipulator` зчитує поточне положення суглоба і безперервно виводить керуючий сигнал в контурі управління `QMotor`. Фактичний розрахунок алгоритму управління сервоприводом міститься в

окремому об'єкту сервоуправління. Клас цього об'єкта є похідним від класу `ServoControl`, який визначає інтерфейс сервоуправління. Сервоуправління за замовчуванням визначено в класі `DefaultPositionControl`, в якому реалізовано контроль положення PID з компенсацією тертя. Класи маніпуляторів типу Puma560 або WAM автоматично створюють екземпляр об'єкта класу `DefaultPositionControl` для зручності програміста. Однак програматор може в будь-який момент переключитися на інше управління сервоприводом.

Для симуляції маніпуляторів необхідна їх динамічна модель. Крім того, для декартового руху необхідна пряма/обернена кінематика та обчислення матриці Якобі.

Всі ці функції розташовані в класах `ManipulatorModel`. Ієрархія класів `ManipulatorModel` відображена на рисунку 4.4.

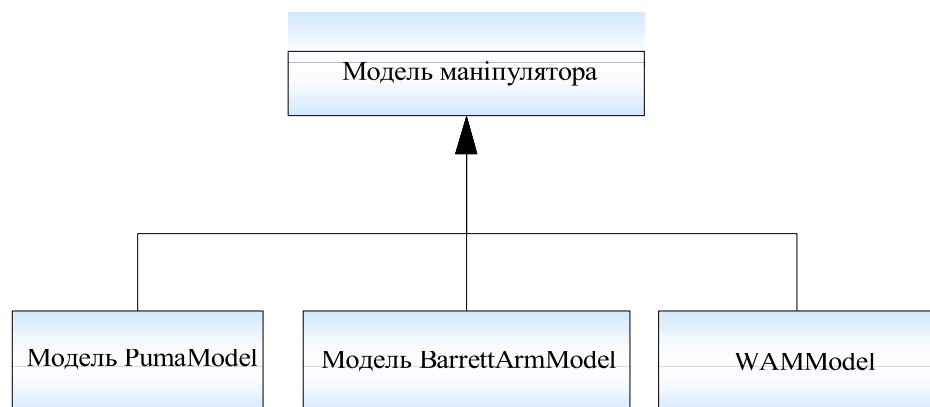


Рисунок 4.4 – Класи моделей маніпулятора

Генерація траєкторії також виконується окремими класами. Клас `TrajectoryGenerator` визначає інтерфейс узагальненого генератора траєкторій. Генератор траєкторій - це будь-який об'єкт, який створює безперервний потік заданих точок і подає цей потік маніпулятору. Маніпулятор викликає функцію `getCurrentSetpoint()` генератора траєкторії для визначення поточного бажаного положення. Також є можливість перемикатися між генераторами з декількома траєкторіями. Клас `QueueTrajectoryGenerator`, який є похідним від класу `TrajectoryGenerator`, є універсальним інтерфейсом генератора траєкторій, який

створює траєкторію вздовж *via* та цільових точок. Клас `DefaultTrajectoryGenerator`, який є похідним від `QueueTrajectoryGenerator`, є специфічною реалізацією генератора траєкторій, який інтерполює як у спільному просторі, так і в декартовому просторі, включаючи змішування шляхів між двома сегментами руху.

4.6 Класи кінцевих ефекторів

Декілька класів роботів відносяться до кінцевих ефекторів, як наведено нижче:

- класи захватів: клас `Gripper` є загальним класом інтерфейсу захвату. Він визначає функції `open()`, `close()` та `relax()`. Клас `DefaultGripper` використовує дві цифрові вихідні лінії для керування захватом, одну цифрову лінію для відкриття захвату та одну для його закриття. Для роботи з `BarrettHand` використовується клас `BarrettHand`;

- класи датчиків сили/крутного моменту: загальний базовий клас `ForceTorqueSensor` визначає інтерфейс датчика сили/крутного моменту. Тобто він визначає функції зчитування зусиль і крутних моментів. Клас `AtiFTSensor` є реалізацією датчика ATI Gamma 30/100 Force/Torque;

- класи зміни інструментів: клас `ToolChanger` є загальним класом інтерфейсу засобу зміни інструментів. У ньому визначаються функції `manipulator` класу `Puma`) через відповідний клас інтерфейсу (наприклад, клас `Manipulator`) за допомогою віртуальних функцій C++.

Отже, загальний код не потрібно змінювати, коли використовується об'єкт іншого класу (наприклад, клас `WAM`), якщо цей об'єкт є похідним від одного класу інтерфейсу. Загальний код дуже корисний для повторного використання коду (наприклад, потрібно записати лише один генератор загальної траєкторії, який можна використовувати з різними типами маніпуляторів). Наступний уривок загального коду є незалежним від маніпулятора кодом, який працює або з роботом `Puma`, або з `WAM`, або з будь-

яким роботом, який буде додано в майбутньому.

```
Manipulator *robot; ObjectManager
om; robot = om.createDerivedObject("leader");
cout << "Current End-Effector Coordinate Frame: "
<< robot->getEndEffectorTransform();
```

Наведений вище код спочатку викликає функцію `createDerivedObject()` для створення об'єкта класів `Puma560`, `BarrettArm` або `WAM`. Потім він керує цим об'єктом за допомогою вказівника на загальний базовий клас (тобто `Manipulator *`). Для того, щоб створити потрібний об'єкт, функція `createDerivedObject()` шукає ім'я об'єкта у глобальному конфігураційному файлі (рисунок 4.3). Потім він зчитує ім'я класу об'єкта з файлу конфігурації та створює об'єкт цього класу. Для цього фреймворк `Robotic Platform` веде список всіх класів програми. Отже, щоб перемкнутися на інший тип маніпулятора, потрібно змінити лише ім'я класу в глобальному файлі конфігурації при використанні загальної програми.

4.7 Модель паралелізму

У той час як для виконання багатьох програмних систем цього часу часто достатньо, роботизовані системи вимагають, щоб такі компоненти, як сервокерування, виконувалися одночасно з іншими компонентами (наприклад, генератором траєкторії). Роботизована платформа виконує всі одночасні завдання на одному ПК в рамках однієї програми керування роботом. Ця програма породжує потоки, якщо потрібне одночасне виконання. Як тільки програма завершується, всі потоки автоматично завершуються. На рис. 4.5 показаний приклад того, як програма користувача породжує кілька потоків.

`lock()`, `unlock()` та `relax()`. Клас `DefaultToolChanger` використовує дві

цифрові вихідні лінії для керування функцією блокування та розблокування пристрою зміни інструментів.

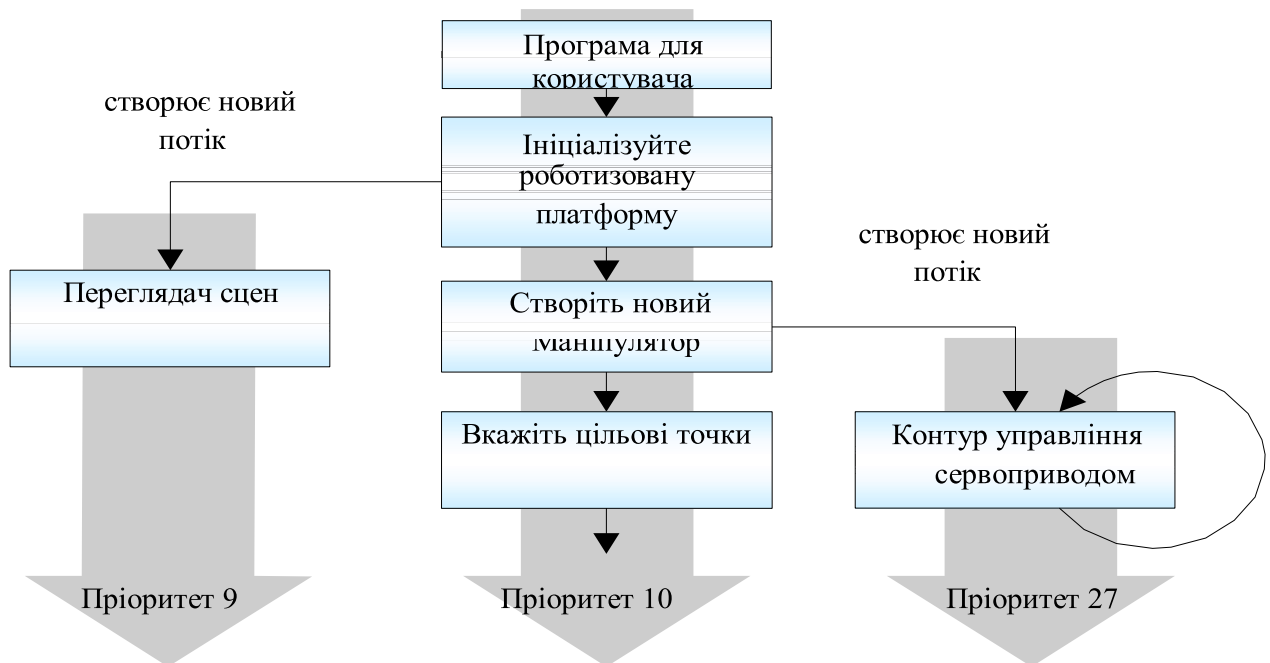


Рисунок 4.5 – Створення нових потоків для паралелізму

4.8 Менеджер об'єктів

У класі `ObjectManager` реалізовано менеджер об'єктів. Кожного разу, коли новий об'єкт створюється в програмі керування роботом користувача, він реєструється в менеджері об'єктів. Аналогічно, кожного разу, коли об'єкт знищується, він видаляється зі списку об'єктів, який ведеться менеджером об'єктів. Менеджер об'єктів містить функціональність для циклічного перегляду цього списку для виконання операцій над кількома об'єктами. Клас `PhysicalObject` рендерить кожен з них, і тим самим, здатний рендерити всю 3D сцену. Функціональність менеджера об'єктів також необхідна для забезпечення загального коду.

Під час запуску програми виконується лише потік 1. При ініціалізації бібліотеки `Robotic Platform` створюється новий потік, який виконує переглядач 3D-сцен. Потім програма управління роботом користувача використовує новий об'єкт класу маніпулятора. Створення цього об'єкта-маніпулятора автоматично породжує третій потік для контуру управління сервоприводом.

Отже, перший потік може йти вперед і вказувати цільові точки для маніпулятора, тоді як контур керування сервоприводом і переглядач сцен працюють у фоновому режимі. Щоб забезпечити поведінку критично важливих за часом завдань у реальному часі, потоки виконуються з різними пріоритетами (наприклад, контур керування сервоприводом виконується з високим пріоритетом 27). Для забезпечення синхронізованого обміну даними між потоками використовується передача повідомлень (що забезпечується класами Client і Server) і стандартні механізми синхронізації потоків (реалізовані в класах Barrier і ReaderWriterLock).

4.9 Реалізація керування за допомогою QMotor

QMotor є повноцінним середовищем для реалізації та налаштування стратегій управління. Для реалізації контуру управління в реальному часі програміст виводить клас з класу ControlProgram і повторно реалізує кілька функцій, які виконують контрольний розрахунок і ведення домашнього господарства. Після того, як програма керування впроваджена та складена, користувач може запустити графічний інтерфейс QMotor, завантажити керуючу програму, запустити її та налаштувати стратегію керування у вікні параметрів керування. Крім того, користувач може відкривати кілька вікон графіка в реальному часі (рисунок 4.6) і встановлювати режими ведення журналу.

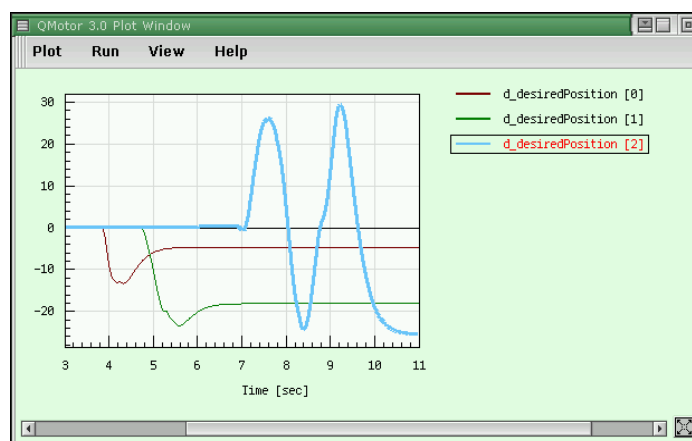


Рисунок 4.6 – Вікно графіка QMotor

Щоб використовувати QMotor для роботизованої платформи, класи, які вимагають циклу керування в реальному часі (наприклад, DefaultManipulator), є похідними від класу ControlProgram QMotor. класи MatrixBase та VectorBase є чистими віртуальними базовими класами, які дозволяють маніпулювати матрицями та векторами невідомого розміру. Матриці та вектори невідомого розміру потрібні під час програмування узагальненого маніпулятора. Клас Transform реалізує однорідну матрицю 4x4, яка зазвичай використовується для представлення рамок координат у роботизованих додатках. Клас MathException використовується для відновлення помилок.

Приклад програми, наведений на рисунку 4.7, виконує поширене завдання в робототехніці: Обчислення рівняння положення. Цей приклад показує, що програмування за допомогою математичної бібліотеки дуже інтуїтивно зрозуміле.

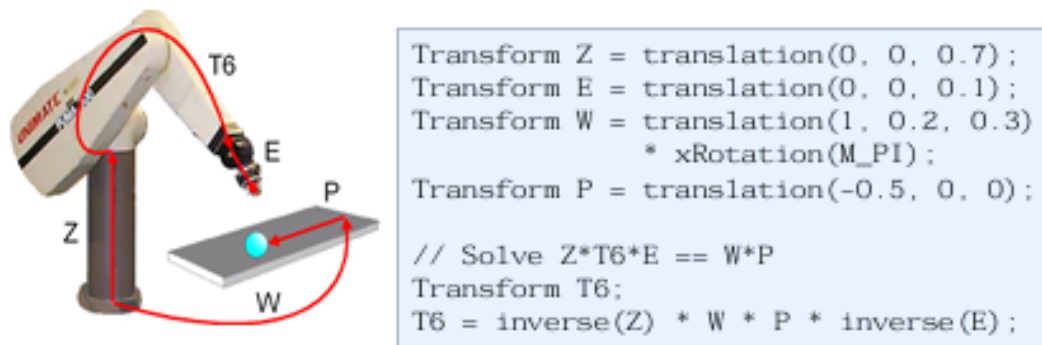


Рисунок 4.7 – Приклад програми для матричних класів

4.10 Математична бібліотека

Попередні бібліотеки управління роботами часто вводили свої специфічні типи даних роботів. Більшість цих типів даних базуються на векторах або матрицях (наприклад, однорідне перетворення є матрицею 4x4). Отже, більш реалістично та гнучко використовувати загальну бібліотеку матриць C++ та визначати роботизовані типи на її основі. Більшість матричних бібліотек, доступних для C++, використовують динамічне виділення пам'яті,

що загрожує втратою детермінованого відгуку в реальному часі [29]. Щоб подолати цей недолік, для роботизованої платформи були розроблені спеціальні класи матриць реального часу (рисунок 4.8), які використовують шаблони для розміру матриці. Отже, розмір матриці відомий на момент компіляції, і динамічний розподіл пам'яті не потрібен. Класи `MatrixBase`, `VectorBase`, `Matrix`, `ColumnVector`, `RowVector` та `Vector` також параметризовані типом даних елементів. Типом даних елемента за замовчуванням є `double`, що є стандартним типом даних з плаваючою комою `Robotics Platform`. Об'єкт

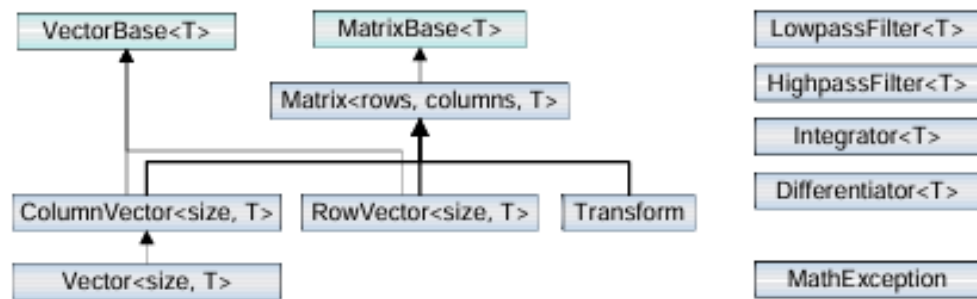


Рисунок 4.8 – Ієрархія класів математичної бібліотеки

Математична бібліотека також надає класи `LowpassFilter` та `HighpassFilter` для числової фільтрації, а також класи `Differentiator` та `Integrator` для чисельного диференціювання та інтегрування. Ці класи параметризовані за допомогою типу даних (тобто працюють зі скалярами, векторами та матрицями).

4.11 Графічний інтерфейс користувача

Щоразу, коли виконується програма роботизованої платформи, відкривається вікно `Scene Viewer`, що відображає 3D-сцену, яка містить усі об'єкти, створені в програмі керування роботом (рисунок 4.9).

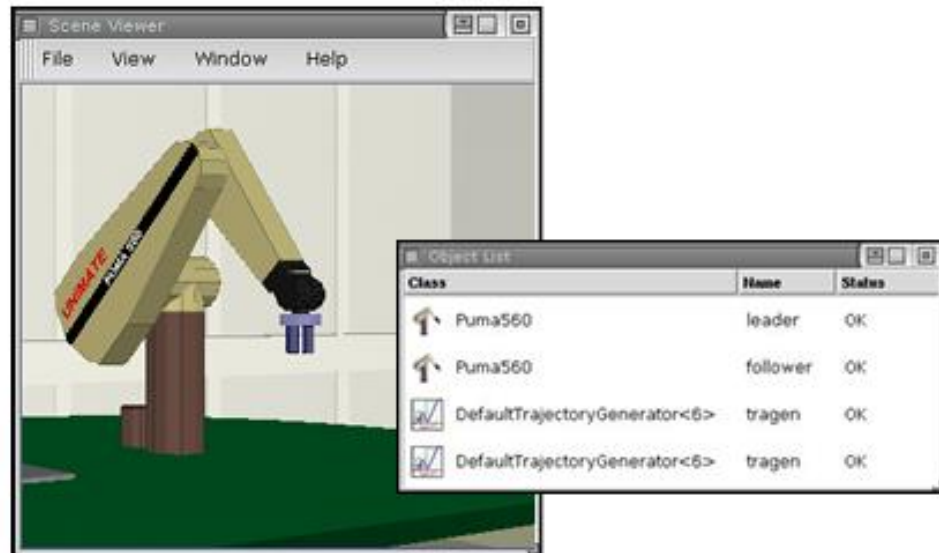


Рисунок 4.9 – Переглядач сцени та вікно списку об'єктів

Щоб зібрати 3D-сцену, Scene Viewer циклічно переглядає всі фізичні об'єкти та отримує їхні 3D-дані Open Inventor. Потім переглядач сцен використовує зв'язки зв'язків між об'єктами для відображення 3D-об'єктів у правильному положенні. Крім того, Scene Viewer постійно оновлює 3D-сцену з поточним станом усіх об'єктів (наприклад, він використовує поточне положення з'єднання маніпулятора для відображення з'єднань у правильному положенні). Отже, 3D-сцена завжди представляє поточний стан апаратного забезпечення (у режимі симуляції представлено змодельований стан апаратного забезпечення). Щоб вибрати найкраще місце для перегляду, користувач може переміщатися в 3D-сцені за допомогою миші. Також користувач може відкрити вікно «Список об'єктів», в якому відображається список всіх об'єктів, які в даний момент створюються програмою управління роботом, включаючи ім'я класу, ім'я об'єкта і статус об'єкта.

Кожен об'єкт має індивідуальне спливаюче меню. Це випадаюче меню з'являється, якщо користувач клацає правою кнопкою миші на об'єкті в області рендерингу Scene Viewer або у вікні Списку об'єктів. У спливаючому меню є опції для вибору режимів 3D-відображення та відкриття панелі керування об'єктом. Крім того, є пункти меню, які є конкретний для класу об'єкта.

Наприклад, об'єкт захвату має додаткові пункти меню для відкриття, закриття та розслаблення захвату.

Robotic Platform надає кілька утиліт з графічним інтерфейсом для калібрування та тестування. Серед іншого, Teachpendant дозволяє користувачеві переміщати маніпулятор в потрібне цільове положення в режимі невагомості і зберігати ці положення в списку. Teachpendant також використовує генератор траєкторії для переміщення маніпулятора назад у збережені положення.

4.12 Написання, компіляція, зв'язок та запуск програм керування роботом

Програма керування роботом спочатку компілюється, а потім зв'язується з бібліотекою Robotic Platform. Після компіляції та зв'язування програми користувач може запустити її з командного рядка. На рисунку 4.10 показано приклад програми керування роботом для простої операції вибору та розміщення.

Кожна програма керування роботом спочатку викликає `RoboticPlatform::init()`. Ця функція ініціалізує платформу та запускає `Scene Viewer`.

```
#include "RoboticPlatform.hpp"

void main(int argc, char *argv[])
{
    RoboticPlatform::init(argc, argv);

    Puma560 puma;
    DefaultGripper gripper;

    DefaultTrajectoryGenerator<6> tragen;
    puma.setTrajectoryGenerator(tragen);
    Transform down = xRotation(M_PI); // End-effector
                                        // pointing down
    gripper.open();
    tragen.moveTo(translation(0, 0.5, 0) * down);
    tragen.stop(1);
    gripper.close();
    tragen.moveTo(translation(0.5, 0.5, 1) * down);
    tragen.moveTo(translation(1, 1, 0) * down);
    tragen.stop(1);
    gripper.open();
}
```

Рисунок 4.10 – Приклад простої програми вибору та розміщення

Потім програма користувача створює всі об'єкти, необхідні для завдання робота (тобто створюються об'єкт захоплення, об'єкт Puma 560 та об'єкт генератора траєкторії). Остання частина прикладу програми використовує об'єкт генератора траєкторії та об'єкт захоплення для переміщення робота до заготовки, закриття захоплення, підняття заготовки та її опускання в цільовому положенні.

4.13 Охорона праці, електробезпека та штучне освітлення

Питання безпеки, зокрема електробезпеки, є невід'ємною частиною сучасних розробок у сфері автоматизації та робототехніки, навіть якщо мова йде про проектування або моделювання робототехнічної системи без її фактичного фізичного впровадження. Незважаючи на те, що дана кваліфікаційна робота носить дослідницький та симуляційний характер, при створенні концепції інтелектуальної системи керування робототехнічною платформою враховувалися принципи безпечного функціонування майбутнього апаратного комплексу в умовах виробничого середовища. З огляду на те, що передбачається використання електронних модулів, сенсорних систем, виконавчих механізмів та елементів живлення, у моделюванні системи управління закладено базові технічні рішення, які відповідають вимогам електробезпеки. До таких відносяться необхідність заземлення струмопровідних частин, передбачення наявності захисних пристроїв у схемах живлення, а також обов'язковий поділ логічного (керуючого) та силового електричного кола.

При формуванні архітектури системи у програмному середовищі були змодельовані варіанти роботи в умовах можливого порушення нормального функціонування — зокрема, враховано сценарії відмови датчиків або втрати живлення. Така передбачуваність потенційно аварійних ситуацій дозволяє інтегрувати до програмної частини алгоритми захисту, які могли б спрацювати у реальній системі – наприклад, аварійну зупинку або

переведення в безпечний режим роботи. Окрему увагу приділено логіці взаємодії платформи з оператором. Навіть в умовах віртуального моделювання важливо було передбачити обмеження на неконтрольований рух, потенційні сценарії перетину небезпечних зон тощо. Подібні умови лягають в основу технічного завдання, і при реалізації фізичної системи вони повинні бути доповнені світловими або звуковими сигналами, захисними бар'єрами, а також системами резервного відключення живлення.

Ще одним важливим фактором, який слід враховувати під час планування використання робототехнічної платформи у виробничому середовищі, є організація штучного освітлення. У контексті робототехнічних систем освітлення має не лише забезпечувати комфортну видимість для обслуговуючого персоналу, але й відіграє роль у стабільності роботи сенсорів, особливо оптичного типу — таких як камери, датчики відстані або контурного розпізнавання. Надмірне затемнення або, навпаки, сильне засвічення можуть викликати помилки в обробці сигналів, що потенційно призводить до неправильних рішень платформи щодо маршруту чи уникнення перешкод. У моделі логіки керування було враховано, що освітлення на виробничій ділянці повинно бути рівномірним і відповідати встановленим нормам для робочих зон, де здійснюється переміщення обладнання. Крім цього, при майбутньому фізичному впровадженні системи доцільно передбачити наявність локального підсвічування в критичних ділянках руху або в зонах завантаження/розвантаження, що підвищить загальну безпеку та точність роботи робототехнічної платформи.

Таким чином, ще на етапі проектування системи керування доцільно враховувати вплив рівня освітленості на функціонування сенсорної системи та алгоритмів розпізнавання. Це дозволить підвищити адаптивність робототехнічного комплексу до різних умов експлуатації без необхідності повної переналаштування апаратної або програмної частини.

Отже, навіть у відсутності реального прототипу, моделювання інтелектуальної системи керування проводилось з урахуванням вимог

електробезпеки, штучного освітлення та загальної концепції охорони праці, яка супроводжує будь-які інженерні рішення в галузі автоматизованих технологій. Врахування таких аспектів на рівні проєктування дозволяє забезпечити подальше масштабування проєкту до реального технічного втілення без необхідності повної переробки логіки системи безпеки. Подібний підхід узгоджується з принципами відповідального інженерного проєктування, особливо в контексті інтеграції роботизованих систем у виробниче середовище [30].

ВИСНОВКИ

Запропоновано розробляти МРП на основі комплексного підходу, який охоплює: методи навігації, способи попередньої обробки даних; сучасні методи та алгоритми інтелектуального управління та нечіткої логіки; сучасна елементна база; методи інтелектуальної обробки та оцінки даних з датчиків в умовах перешкод і неповноти інформації; методи та засоби автоматизованого проектування апаратних і програмних засобів.

При розробці ІСУР для МРР обрані такі принципи: ієрархія; систематичний; варіативний склад обладнання; Модульність; відкритість програмного забезпечення та максимальне використання стандартних драйверів та програмного забезпечення; спеціалізація та адаптація апаратних і програмних засобів до структури алгоритмів обробки та формування команд управління.

Розроблена структура ІСУР для колісних МРП, основними компонентами яких є: вузли дистанційного керування, інтелектуальні автономні та інтелектуальні гібридні навігаційні блоки, засоби збору, зберігання та обробки даних про рух з датчиків, канали передачі відео та даних управління.

Розроблено звід правил для системи керування нечіткою логікою на базі ІСУР.

Наведено результати моделювання руху колісного МРП з різними значеннями показань датчиків у навколишньому середовищі на основі розроблених правил.

Запропоновано виконувати програмну розробку ІСУР для МРЦ у віддаленому режимі.

Такий підхід дозволяє швидко розробляти і налагоджувати програмне забезпечення системи управління без необхідності постійно перевантажувати програмне забезпечення на мобільну платформу.

Випробувана затримка передачі даних близько 50 мс розробленого бездротового датчика відстані повинна вважатися прийнятною для реалізації процесу віддаленого налагодження програмного забезпечення.

На відміну від попередніх розподілених платформ керування роботами, розроблена представляє однорідну, нерозподілену об'єктно-орієнтовану архітектуру. Тобто, на основі технології ПК та QNX RTP, усі компоненти реального часу та нереального часу інтегровані в єдину бібліотеку C++. Архітектура забезпечує ефективну інтеграцію та розширюваність пристроїв, стратегій керування, генерації траєкторій та компонентів графічного інтерфейсу. Платформу побудовано на середовищі керування QMotor для реєстрації даних, налаштування параметрів керування та побудови графіків у реальному часі. Нова математична бібліотека реального часу спрощує операції та забезпечує простий у використанні інтерфейс програмування.

Вбудовані компоненти графічного інтерфейсу, такі як Scene Viewer та панелі керування, забезпечують комфортну роботу та швидкий час запуску навіть для користувачів, які не мають досвіду в програмуванні на C++.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Malik PK, Sharma R, Singh R, et al. Industrial internet of things and its applications in industry 4.0: State of the art. *Computer Communications* 2021; 166: 125-139.
2. Seeja G, Reddy O, Kumar KVR, et al. Internet of things and robotic applications in the industrial automation process. *Innovations in the Industrial Internet of Things (IIoT) and Smart Factory*. IGI Global, 2021.p.50-64.
3. Munirathinam S. Industry 4.0: Industrial internet of things (IIOT). *Advances in Computers*. Elsevier, 2020.p.129-164.
4. Методичні вказівки з підготовки кваліфікаційної роботи для здобувачів першого (бакалаврського) рівня вищої освіти денної і заочної форми навчання спеціальності 151 – Автоматизація та комп'ютерно-інтегровані технології освітньої [Електронний ресурс] / упоряд.: І. Ш. Невлюдов, О. І. Филипенко, О. В. Токарева, С. П. Новоселов, О.В Сичова ; М-во освіти і науки України, Харків. нац. ун-т радіоелектроніки. – електрон. вид. – Харків : ХНУРЕ, 2023. – 760
5. ДСТУ 3008-15. Документація. Звіти у сфері науки та техніки. структура та правила оформлення. Введ. 2015-06-22. К. Держстандарт України, 2017. 29 с.
6. Liu L, Song W, Liu Y. Leveraging digital capabilities toward a circular economy: Reinforcing sustainable supply chain management with Industry 4.0 technologies. *Computers & Industrial Engineering* 2023; 178: 109113.
7. Kalsoom T, Ramzan N, Ahmed S, et al. Advances in sensor technologies in the era of smart factory and industry 4.0. *Sensors* 2020; 20: 6783.
8. Katterpur A, Dey S, Balamuralidhar P. Knowledge based hierarchical decomposition of industry 4.0 robotic automation tasks. In: *IECON 2018-44th Annual Conference of the IEEE Industrial Electronics Society*. 2018.p.3665-3672.
9. Zhong RY, Xu X, Klotz E, et al. Intelligent manufacturing in the context

of industry 4.0: a review. *Engineering* 2017; 3: 616-630.

10. Ammar M, Haleem A, Javaid M, et al. Improving material quality management and manufacturing organizations system through Industry 4.0 technologies. *Materials Today: Proceedings* 2021; 45: 5089-5096.

11. Sindhwani N, Anand R, George AS, et al. *Robotics and Automation in Industry 4.0: Smart Industries and Intelligent Technologies*. CRC Press, 2024.

12. Yang Y, Pan W. Automated guided vehicles in modular integrated construction: Potentials and future directions. *Construction Innovation* 2021; 21: 85-104.

13. Villani V, Pini F, Leali F, et al. Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications. *Mechatronics* 2018; 55: 248-266.

14. Singh Rajawat A, Bedi P, Goyal S, et al. Reformist framework for improving human security for mobile robots in industry 4.0. *Mobile Information Systems* 2021; 2021: 1-10.

15. Jamwal A, Agrawal R, Sharma M, et al. Industry 4.0 technologies for manufacturing sustainability: a systematic review and future research directions. *Applied Sciences* 2021; 11: 5725.

16. Berx N, DecréW, Morag I, et al. Identification and classification of risk factors for human-robot collaboration from a system-wide perspective. *Computers & Industrial Engineering* 2022; 163: 107827.

17. Jardine A. K. S., Lin D., Banjevic D. *A review on machinery diagnostics and prognostics implementing condition-based maintenance*. Mechanical Systems and Signal Processing, 2006.

18. Klingenberg CO, Borges MAV, Antunes Jr JAV. Industry 4.0 as a data-driven paradigm: a systematic literature review on technologies. *Journal of Manufacturing Technology Management* 2021; 32: 570-592.

19. Li C, Chen Y, Shang Y. A review of industrial big data for decision making in intelligent manufacturing. *Engineering Science and Technology, an International Journal* 2022; 29: 101021.

20. A. Aniculaesei, J. Grieser, A. Rausch, K. Rehfeldt and T. Warnecke, Toward a Holistic Software Systems Engineering Approach for Dependable Autonomous Systems, in: Proceedings of the IEEE/ACM 1st International Workshop on Software Engineering for AI in Autonomous Systems (SEFAIAS), Gothenburg, Sweden, 2018, pp. 23-30.

21. S. Wang, G. Xu, T. Liu and Y. Zhu, Robust Real-time Obstacle Avoidance of Wheeled Mobile Robot based on Multi-Sensor Data Fusion, in: Proceedings of the IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chongqing, China, 2021, pp. 2383-2387. doi: 10.1109/IAEAC50856.2021.9391021.

22. A. Medina-Santiago, L.A. Morales-Rosales, C.A. Hernández-Gracidas, I. Algreto-Badillo, A.D. Pano Azucena, J.A. Orozco Torres, Reactive Obstacle-Avoidance Systems for Wheeled Mobile Robots Based on Artificial Intelligence. *Applied Sciences* (2021) 11(14):6468.

23. C.L.P. Chen, D. Yu, L. Liu. Automatic leader-follower persistent formation control for autonomous surface vehicles. *IEEE Access* (2019) 7, 12146–12155.

24. I. Tsmots, R. Tkachenko, V. Teslyuk, Y. Opotyak and V. Rabyk, Hardware Components for Nonlinear Neuro-like Data Protection in Mobile Smart Systems, in: Proceedings of the IEEE 17th International Conference on Computer Sciences and Information Technologies (CSIT), Lviv, Ukraine, 2022, pp. 198-202. doi: 10.1109/CSIT56902.2022.10000636.

25. D. Hutabarat, M. Rivai, D. Purwanto and H. Hutomo, LIDAR-based obstacle avoidance for the autonomous mobile robot, in: Proceedings of the 12th International Conference on Information & Communication Technology and System (ICTS), Surabaya, Indonesia, 2019, pp. 197-202. doi: 10.1109/ICTS.2019.8850952.

26. B. Fares, H. Souifi, M. Ghribi and Y. Bouslimani, Omnidirectional Platform for Autonomous Mobile Industrial Robot, in: Proceedings of the IEEE 3rd Eurasia Conference on IOT, Communication and Engineering (ECICE), Yunlin,

Taiwan, 2021, pp. 633-638. doi: 10.1109/ECICE52819.2021.9645621.

27. P. Yue, J. Xin, H. Zhao, D. Liu, M. Shan and J. Zhang, Experimental Research on Deep Reinforcement Learning in Autonomous navigation of Mobile Robot, in: Proceedings of the 14th IEEE Conference on Industrial Electronics and Applications (ICIEA), Xi'an, China, 2019, pp. 1612-1616. doi: 10.1109/ICIEA.2019.8833968.

28. S. Noh, J. Park and J. Park, Autonomous Mobile Robot Navigation in Indoor Environments: Mapping, Localization, and Planning, in: Proceedings of the International Conference on Information and Communication Technology Convergence (ICTC), Jeju, Korea (South), 2020, pp. 908-913. doi: 10.1109/ICTC49870.2020.9289333.

29. F. Wildani, R. Mardiaty, E. Mulyana, A. E. Setiawan, R. R. Nurmalasari and N. Sartika, Fuzzy Logic Control for Semi-Autonomous Navigation Robot Using Integrated Remote Control, in: Proceedings of the 8th International Conference on Wireless and Telematics (ICWT), Yogyakarta, Indonesia, 2022, pp. 1-5. doi: 10.1109/ICWT55831.2022.9935458.

30. Комплекс навчально-методичного забезпечення навчальної дисципліни «Безпека праці в індустрії ІТ-технологій» підготовки освітнього рівня бакалавр усіх спеціальностей та усіх напрямів університету / ХНУРЕ; розроб.: Т. Є. Стиценко, Г. В. Пронюк, Н. М. Сердюк. – Харків, 2017. – 122 с.