

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Автоматики і комп'ютеризованих технологій
(повна назва)
Кафедра Комп'ютерно-інтегрованих технологій, автоматизації та
робототехніки
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)
«Розроблення системи керування групою мобільних роботів
у умовах невизначеності»
(тема)

Виконав:
здобувач 2 року навчання,
групи КТРСм-24-2
Максим МОІСЄЄВ
(власне ім'я, прізвище)

Спеціальність 174 Автоматизація,
комп'ютерно-інтегровані технології
та робототехніка
(код і повна назва спеціальності)
Тип програми освітньо-професійна
Освітня програма Комп'ютеризовані
та робототехнічні системи
(повна назва освітньої програми)

Керівник проф. Владислав ЄВСЄЄВ
(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри КІТАР

Ігор НЕВЛЮДОВ
(підпис) (власне ім'я, прізвище)

2025 р.

Харківський національний університет радіоелектроніки

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

ЗАТВЕРДЖУЮ:

Зав. кафедри

(підпис)

«16» грудня 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Моїсєєву Максиму Сергійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи: «Розроблення системи керування групою мобільних роботів
у умовах невизначеності»

затверджена наказом університету від 10.11.2025 р. № 1018 Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 16.12.2025 р.

3. Вихідні дані до роботи 3.1 Платформа 4WD; 3.2 Двигуни DC 6 В;
3.3 Драйвер двигунів L298N; 3.4 Система керування на базі Raspberі Pi4 та
камера PiCam; 3.5 2 ультразвукових датчика (HC-04); 3.6 3-х ланковий
маніпулятор (серводвигуни 90 g) та захватний пристрій

4. Перелік питань, що потрібно опрацювати в роботі 4.1 Вступ; 4.2 Аналіз
колаборативних роботів-маніпуляторів та систем керування; 4.3 Формування
вимог до математичної моделі; 4.4 Розробка математичної моделі опису
параметрів мобільного робота; 4.5 Розробка математичного представлення
навколишньої середовища; 4.6 Розробка математичної моделі прийняття рішень
групою мобільних роботів на базі нейронної мережі; 4.7 Розробка метода
керування групою мобільних роботів; 4.8 Вибір мови та середовища розробки;
4.9 Розробка алгоритму роботи програмного модуля; 4.10 Реалізація нейронної
мережі; 4.11 Проведення моделювання та аналіз отриманих результатів;
4.12 Висновки

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

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Аналіз колаборативних роботів-маніпуляторів та систем керування	02.09.2025-15.09.2025	виконано
2	Формування вимог до математичної моделі	16.09.2025-26.09.2025	виконано
3	Розробка математичної моделі опису параметрів мобільного робота	27.09.2025-05.10.2025	виконано
4	Розробка математичного представлення навкол. середовища	06.10.2025-11.10.2025	виконано
5	Розробка математичної моделі прийняття рішень групою мобільних роботів на базі нейронної мережі	12.10.2025-24.10.2025	виконано
6	Розробка метода керування групою мобільних роботів	25.10.2025-08.11.2025	виконано
7	Вибір мови та середовища розробки	09.11.2025-10.11.2025	виконано
8	Розробка алгоритму роботи програмного модуля	11.11.2025-13.11.2025	виконано
9	Реалізація нейронної мережі	14.11.2025-25.11.2025	виконано
10	Проведення моделювання та аналіз отриманих результатів	26.11.2025-02.12.2025	виконано
11	Оформлення пояснювальної записки	03.12.2025-07.12.2025	виконано

(підпис)

(посада, власне ім'я, прізвище)

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

" 15 " грудня 2025 р.



Максим МОІСЄЄВ

РЕФЕРАТ

Пояснювальна записка: 84 с., 2 табл., 10 рис., 3 дод., 18 джерел.

ГРУПОВЕ КЕРУВАННЯ МОБІЛЬНИМИ РОБОТАМИ, ПЛАНУВАННЯ МАРШРУТУ, МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ, НЕЙРОННІ МЕРЕЖІ, МУЛЬТИАГЕНТНІ СИСТЕМИ

Метою роботи є підвищення ефективності побудови маршруту переміщення колаборативних роботів при груповому керуванні.

Об'єктом дослідження є процес побудови маршруту для колаборативних роботів в умовах групового керування.

Предметом дослідження є моделі, методи та програмне забезпечення для численного моделювання маршруту переміщення групи колаборативних роботів.

В кваліфікаційній роботі було проведено аналіз сучасних мобільних роботів. Проаналізовано математичні моделі представлення мобільних роботів та методи групового керування мобільними роботами, а також нейронні мережі. Сформовано вимоги до математичної моделі, розроблено математичну модель опису параметрів мобільного робота та математичне представлення навколишнього середовища. Розроблено математичну модель прийняття рішень групою мобільних роботів на базі нейронної мережі та метод керування групою мобільних роботів. Проведено вибір мови та середовища розробки. Розроблено алгоритм роботи програмного модуля та реалізовано нейронну мережу. Проведено моделювання та аналіз отриманих результатів.

Отримані результати роботи можна віднести до Цілі сталого розвитку 9 «Промисловість, інновації та інфраструктура», а саме 9.5. Створити фінансову та інституційну системи (інноваційну інфраструктуру), що забезпечуватимуть розвиток наукових досліджень та науковотехнічних (експериментальних) розробок.

ABSTRACT

Explanatory note: 84 p., 3 table, 10 fig., 3 app, 18 sources.

GROUP CONTROL OF MOBILE ROBOTS, ROUTE PLANNING, MATHEMATICAL MODELING, NEURAL NETWORKS, MULTIAGENT SYSTEMS

The purpose of the work is to increase the efficiency of constructing the route of movement of collaborative robots in group control.

The object of the study is the process of constructing the route for collaborative robots in group control conditions.

The subject of the study is models, methods and software for numerous modeling of the route of movement of a group of collaborative robots.

The qualification work analyzed modern mobile robots. Mathematical models of the representation of mobile robots and methods of group control of mobile robots, as well as neural networks, were analyzed. Requirements for the mathematical model were formed, a mathematical model for describing the parameters of a mobile robot and a mathematical representation of the environment were developed. A mathematical model of decision-making by a group of mobile robots based on a neural network and a method for controlling a group of mobile robots were developed. The language and development environment were selected. The algorithm for the software module was developed and the neural network was implemented. Modeling and analysis of the results were carried out.

The results of the work can be attributed to Sustainable Development Goal 9 "Industry, Innovation and Infrastructure", namely 9.5. Create a financial and institutional system (innovation infrastructure) that will ensure the development of scientific research and scientific and technical (experimental) developments.

ЗМІСТ

Перелік умовних скорочень	9
Вступ.....	10
1 Аналіз колаборативних роботів-маніпуляторів та систем керування	12
1.1 Аналіз сучасних мобільних роботів	12
1.2 Аналіз математичних моделей представлення мобільних роботів.....	14
1.3 Аналіз методів групового керування мобільними роботами	16
1.4 Аналіз нейронних мереж	18
1.5 Постановка задач досліджень	20
2 Розробка математичної моделі формалізації мобільного робота.....	22
2.1 Формування вимог до математичної моделі	22
2.2 Розробка математичної моделі опису параметрів мобільного робота	24
2.3 Розробка математичного представлення навколишньої середовища.....	36
2.4 Висновки до 2 розділу	40
3 Розробка метода керування групою мобільних роботів	41
3.1 Розробка математичної моделі прийняття рішень групою мобільних роботів на базі нейронної мережі	41
3.2 Розробка метода керування групою мобільних роботів	50
3.3 Висновки до 3 розділу	54
4 Розробка програми моделювання керування групою мобільних роботів.....	55
4.1 Вибір мови та середовища розробки.....	55
4.2 Розробка алгоритму роботи програмного модуля	56
4.3 Реалізація нейронної мережі	60
4.4 Проведення моделювання та аналіз отриманих результатів	63
4.5 Охорона праці	67
4.6 Висновки до 4 розділу	68
Висновки	70
Перелік джерел посилання	72
Додаток А Код програми моделювання.....	75

Додаток Б Апробація результатів кваліфікаційної роботи	81
Додаток В Демонстраційний матеріал	83

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

TЗ – технічне завдання;

CNN – згорткова нейронна мережа;

GNN – графова нейронна мережа;

MLP – багатошарова повнозв’язна мережа / багатошаровий перцептрон;

ReLU – rectified linear unit;

RL – навчання з підкріпленням;

SLAM – картографування та одночасна локалізація;

MARL – мультиагентне навчання;

Occupancy grids – матриці зайнятості;

Hidden layer – прихований шар.

ВСТУП

Розвиток Індустрії 5.0 та активне впровадження роботизованих систем у виробництво, логістику, сервісні технології та інші сфери зумовлюють необхідність створення ефективних методів групового керування мобільними роботами. На відміну від традиційних автономних систем, колаборативні роботи мають функціонувати у спільному робочому просторі з людиною та іншими роботами, що висуває підвищені вимоги до їхньої безпеки, адаптивності й передбачуваності поведінки. Особливої актуальності набуває задача побудови оптимальних та безпечних маршрутів для групи роботів у умовах невизначеності, коли середовище може змінюватися, а параметри задачі – варіюватися в режимі реального часу.

Більшість сучасних методів навігації орієнтовані переважно на керування окремим роботом у динамічному середовищі, що не враховує складності взаємодії між декількома автономними агентами. У таких умовах традиційні алгоритми не гарантують уникнення колізій, узгодженості маршрутів та виконання спільної цілі групою роботів. Тому виникає потреба у розробленні нових моделей і методів, здатних забезпечити колективну поведінку, координацію та безпечний рух мобільних роботів за різних сценаріїв.

Мета роботи – підвищення ефективності побудови маршруту переміщення колаборативних роботів при груповому керуванні.

Об'єкт дослідження – процес побудови маршруту для колаборативних роботів в умовах групового керування.

Предмет дослідження – моделі, методи та програмне забезпечення для численного моделювання маршруту переміщення групи колаборативних роботів.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- провести аналіз сучасних мобільних роботів;
- провести аналіз математичних моделей представлення мобільних роботів;

- провести аналіз методів групового керування мобільними роботами;
- провести аналіз нейронних мереж;
- провести аналіз формування вимог до математичної моделі;
- розробити математичну модель опису параметрів мобільного робота;
- розробити математичне представлення навколишнього середовища;
- розробити математичну модель прийняття рішень групою мобільних роботів на базі нейронної мережі;
- розробити метод керування групою мобільних роботів;
- провести вибір мови та середовища розробки;
- розробити алгоритм роботи програмного модуля;
- реалізувати нейронну мережу;
- провести моделювання та аналіз отриманих результатів.

Кваліфікаційна робота виконана згідно ДСТУ 3008–15 [1], методичних вказівок [2], а робота отримала апробацію у [3].

1 АНАЛІЗ КОЛАБОРАТИВНИХ РОБОТІВ-МАНІПУЛЯТОРІВ ТА СИСТЕМ КЕРУВАННЯ

1.1 Аналіз сучасних мобільних роботів

Сучасні мобільні роботи є ключовим елементом робототехнічних систем, що застосовуються у промисловості, логістиці, сільському господарстві, сфері послуг та обороні. Їхньою характерною особливістю є здатність самостійно переміщатися у просторі, взаємодіяти з навколишнім середовищем та виконувати поставлені завдання з високим рівнем автономності. За конструктивними ознаками мобільні роботи розділяються на декілька основних типів: колісні, гусеничні, крокуючі та комбіновані системи. Колісні платформи є найпоширенішими завдяки високій швидкості руху, простоті конструкції та енергоефективності. Їх використовують у внутрішньоскладських логістичних системах, транспортних роботах та сервісних пристроях. Гусеничні роботи мають підвищену прохідність, що робить їх придатними для застосування у складних умовах місцевості або у зонах, де поверхня є нерівною або нестабільною. Крокуючі роботи мають найвищу мобільність у контексті подолання перешкод, однак їх конструкція є складнішою та енергозатратнішою. В свою чергу комбіновані мобільні роботи об'єднують переваги декількох типів рухів і застосовуються у спеціалізованих задачах, наприклад, для інспекції важкодоступних об'єктів [4].

Функціональні можливості сучасних мобільних роботів визначаються комплексом сенсорів, що забезпечують їхню «обізнаність» про навколишнє середовище. До найбільш поширених сенсорних засобів належать лідари, камери (RGB, стерео та глибинні), ультразвукові та інфрачервоні датчики, GPS-модулі, інерційні вимірювальні блоки та радары, комбінація різних типів сенсорів дозволяє роботам виконувати локалізацію та картографування (SLAM), уникати перешкод, оцінювати стан поверхні, розпізнавати об'єкти та інших агентів у спільній робочій зоні.

Одним із найважливіших аспектів сучасних мобільних роботів є рівень автономності, який визначається здатністю до самостійного планування маршруту, адаптації до змін умов та прийняття оптимальних рішень у реальному часі. Використання алгоритмів штучного інтелекту, зокрема нейронних мереж та методів навчання з підкріпленням, дозволяє створювати роботів, здатних працювати в умовах невизначеності та змінюваного середовища. Сучасні роботи не лише реагують на зовнішні фактори, а й прогнозують їх розвиток, що підвищує стабільність руху та ефективність виконання завдань [5].

Окреме місце займають мобільні роботи, що працюють у групах, такі системи застосовуються у логістиці (координації роботизованих складів), у військових застосуваннях (групи безпілотних платформ), у дослідженні середовища (рої дронів) чи у виробничих процесах. Основною перевагою групового підходу є підвищення ефективності виконання колективних задач, таких як паралельний огляд території, транспортування вантажів або колективне картографування, однак ускладнюється система керування мобільних роботів, оскільки виникає потреба у синхронізації дій, запобіганні зіткненням, організації обміну інформацією та забезпеченні узгодженості рішень між агентами.

Ще одним немало важливим критерієм аналізу є програмно-алгоритмічне забезпечення мобільних роботів. Значна частина таких систем працює на основі ROS (Robot Operating System), що забезпечує гнучкість, модульність та доступ до численних бібліотек для навігації, планування маршруту та взаємодії з сенсорами, завдяки чому мобільні роботи можуть бути легко інтегровані у більш широкі автоматизовані комплекси [6].

Підсумовуючи, можна сказати, що сучасні мобільні роботи характеризуються високим рівнем технологічної складності, широким спектром конструктивних рішень та значною функціональністю. Їх розвиток рухається у напрямі підвищення автономності, точності навігації, здатності працювати у невизначених умовах та ефективно взаємодіяти у групі. Проведений аналіз дозволяє визначити вимоги до моделей та методів керування, які необхідні для створення ефективної системи планування маршрутів групи мобільних роботів.

1.2 Аналіз математичних моделей представлення мобільних роботів

Аналіз математичних моделей представлення мобільних роботів є важливим етапом при розробленні систем навігації, керування та прогнозування поведінки робототехнічних мобільних платформ. Математична модель визначає спосіб опису руху, взаємодії з навколишнім середовищем та динаміки робота, і саме від її вибору залежить коректність подальшого планування траєкторії та координації дій у групі роботів. Розглянемо кілька розроблених підходів до моделювання мобільних платформ, які відрізняються рівнем точності, складністю та можливостями у застосуванні в умовах невизначеності.

Одним із найпоширеніших підходів є кінематичні моделі, що описують рух робота у вигляді залежності між бажаними швидкостями та зміною координат у просторі. Ці моделі є простими, обчислювально ефективними та придатними для роботів із неголономними обмеженнями, наприклад диференціально-керованих платформ або роботів із приводом на двох незалежних колесах. Кінематичні моделі широко використовуються у задачах високорівневого планування маршруту, де достатньо знати загальну траєкторію без урахування силових та інерційних впливів, проте ці моделі мають суттєве обмеження, так як вони не враховують динаміку руху, тому у випадках, коли важливими є прискорення, маса платформи або вплив зовнішніх сил, їх застосування є недостатнім [7].

На відміну від кінематичного підходу, динамічні моделі забезпечують більш точний опис поведінки мобільного робота, враховуючи сили, моменти, прискорення, інерцію та реальні обмеження приводів. Такі моделі базуються на законах Ньютона–Ейлера або на методі Лагранжа, що дозволяє враховувати енергетичні параметри системи. Динамічні моделі широко застосовуються у роботах, де необхідно точно контролювати швидкісні режими, забезпечувати плавне гальмування або уникати різких маневрів, що можуть призвести до нестабільності. Попри високу точність, динамічні моделі є значно складнішими, а їх використання потребує значних обчислювальних ресурсів, що ускладнює застосування у реальному часі, особливо у групових системах із великою кількістю агентів.

Як альтернативу класичним рівнянням руху застосовують моделі станів, що описують мобільного робота у вигляді векторів стану та керування, такий підхід дозволяє лаконічно представити систему у формі рівнянь у просторі станів, що є зручним для застосування методів оптимального керування, фільтрації та прогнозування. Моделі станів добре працюють у поєднанні з алгоритмами локалізації, такими як фільтр Калмана або його нелінійні варіанти, однак ефективність моделей станів значною мірою залежить від правильності вибору параметрів системи та наявності достовірних сенсорних даних, що може бути проблемою у випадках шумних або нестабільних вимірювань [8].

У складних та невизначених середовищах зростає популярність моделей на основі нейронних мереж, які здатні апроксимувати динаміку мобільних роботів без необхідності формального виведення фізичних рівнянь. Такі моделі можуть навчатися на реальних даних, адаптуючись до різних умов експлуатації та непередбачуваних впливів. Нейронні мережі дозволяють описувати систему як нелінійну функцію входів і виходів, що робить їх ефективними у задачах прогнозування руху, прийняття рішень та планування траєкторії, насамперед, їхнім недоліком є залежність від великої кількості навчальних даних, складність інтерпретації чи можливість помилок у ситуаціях, що виходитимуть за межі навчального набору.

Для групових систем мобільних роботів великого значення набувають моделі на основі графів, що вважають кожного робота як окремого агента, а взаємодію між ними описують через графову структуру. Такі моделі дозволяють врахувати локальні зв'язки між агентами, обмін інформацією, формування колективної поведінки та адаптацію до змін кількості роботів, а у поєднанні з графовими нейронними мережами такі моделі забезпечують високий рівень масштабованості та ефективності при колективному ухваленні рішень. Недоліком даного методу є складність реалізації та значні вимоги до обчислювальних ресурсів.

Кожен з описаних вище типів математичної моделі має свої переваги та обмеження, кінематичні моделі забезпечують простоту та швидкість, але обмежені у точності, динамічні моделі дають реалістичний опис, проте

ускладнюють обчислення, моделі станів є універсальними, але чутливими до якості даних, нейромережеві підходи забезпечують адаптивність, однак потребують навчання, а графові моделі є найкращими для групових робіт, але коштовні у реалізації. В свою чергу, вибір моделі залежить від конкретної задачі, для високорівневого планування буде достатньо кінематики, для точного керування потрібна буде динаміка, а для групових робіт оптимальними є графові та нейромережеві підходи [9].

1.3 Аналіз методів групового керування мобільними роботами

У сучасних підходах простежується перехід від централізованих систем до децентралізованих, що дозволяє зменшити залежність від єдиного вузла, підвищити надійність і масштабованість, а також забезпечити стійкість до відмов окремих роботів. Централізовані методи заключаються в тому, що у таких системах весь процес прийняття рішень концентрується в одному керуючому модулі, який збирає інформацію з датчиків усіх роботів, обчислює оптимальне розподілення цілей та командує виконанням дій. Їхня перевага полягає у глобальному баченні стану системи та можливості оптимізувати загальний критерій, але водночас вони демонструють низьку стійкість до збоїв і слабку адаптивність у динамічних середовищах, де обмін даними може бути нестабільним.

Децентралізовані методи, на відміну від централізованих, покладаються на локальні рішення кожного робота, кожен агент формує власну модель середовища і взаємодіє з іншими роботами через локальні правила, що дозволяє формувати узгоджену поведінку без глобального контролера. Одним із найпоширеніших підходів є поведінкові методи, засновані на принципах самоорганізації, де роботи виконують прості локальні дії – уникнення зіткнень, вирівнювання напрямку руху та притягання до групи – що в результаті створює колективний ефект. Ці системи добре адаптуються до непередбачуваних умов і змін у складі групи, проте складно гарантують оптимальність маршруту або швидкість розв'язання задачі [10].

Методи на основі потенційних полів дозволяють розглядати групу роботів як систему частинок, що взаємодіють у штучно сформованих силових полях. Ці поля можуть притягувати до цілі та відштовхувати від перешкод і інших роботів. Метод вирізняється простотою реалізації та плавністю руху, однак має недолік у вигляді можливості потрапляння агентів у локальні мінімуми, що ускладнює застосування у складних середовищах, схожих на лабіринти.

Оптимізаційні методи, зокрема еволюційні алгоритми та рійові оптимізації, застосовуються для пошуку найкращого розподілу завдань або побудови оптимальних траєкторій у групі. Вони демонструють високу ефективність у задачах з великою кількістю параметрів, де класичні методи не дають результатів. Проте час обчислень може бути значним, а результат – не завжди гарантовано оптимальним, оскільки стохастична природа таких алгоритмів передбачає ймовірнісний характер рішення [11].

У сучасних робототехнічних системах активно розвиваються методи групового керування на основі мультиагентних систем, де кожен робот розглядається як інтелектуальний агент зі здатністю до локального планування, сприйняття та комунікації. Такі системи поєднують елементи поведінкових правил, оптимізаційних стратегій та машинного навчання, дозволяючи досягати високої автономності. Використання навчання з підкріпленням забезпечує можливість спільного набуття досвіду, коли роботи покращують свої стратегії взаємодії через накопичення статистики успішних та неуспішних дій. Проте такі системи потребують великих обсягів навчальних даних та обчислювальних ресурсів, а також складні у валідації в реальних умовах.

З-поміж гібридних методів варто виділити підходи, що поєднують централізоване планування високого рівня з децентралізованим виконанням локальних маневрів, забезпечуючи баланс між глобальною оптимальністю та локальною адаптивністю. У таких системах центральний модуль може задавати загальні цілі та стратегічні напрямки руху, тоді як кожен робот самостійно ухвалює тактичні рішення, враховуючи локальні перешкоди й сигнали інших роботів. Це підвищує стійкість групи до змін середовища та відмов окремих агентів.

Проведений аналіз методів групового керування демонструє, що сучасні тенденції рухаються до децентралізації, адаптивності та використання інтелектуальних алгоритмів, здатних враховувати невизначеність та складність реальних середовищ, вибір методу залежить від конкретної задачі, динаміки середовища, вимог до швидкості реакції та рівня автономності роботів [12].

1.4 Аналіз нейронних мереж

У сучасних системах керування мобільними роботами нейронні мережі відіграють важливу роль, забезпечуючи можливість роботи в умовах невизначеності, адаптивного планування маршруту та прийняття рішень у багатокомпонентних динамічних середовищах. В залежності від архітектури, нейронні мережі по-різному обробляють інформацію, мають різні можливості щодо урахування просторово-часових залежностей та демонструють різну ефективність у мультиагентних системах, через що виникає необхідність аналізу різних типів нейронних мереж для визначення найбільш доцільних підходів у задачах керування групою мобільних роботів.

Першими розглянемо багатосарові перцептрони (MLP), які є базовими повнозв'язними архітектурами, вони характеризуються простотою реалізації та високою швидкістю навчання, проте не здатні враховувати часові або просторові залежності. MLP доцільно використовувати лише для допоміжних задач, таких як обробка табличних даних або апроксимація функцій, але вони є недостатньо ефективними для прогнозування траєкторій чи колективної взаємодії роботів.

Наступними є згорткові нейронні мережі (CNN) демонструють високу ефективність при роботі з просторовими даними, такими як карти місцевості, матриці зайнятості (occupancy grids) чи зображення з камер. Завдяки механізму згортки CNN уміють виявляти локальні патерни та ознаки, що робить їх придатними для навігаційних задач, пов'язаних із виявленням перешкод або інтерпретацією навколишнього середовища. Проте CNN практично не обробляють часових залежностей, що суттєво обмежує їх застосування у динамічних середовищах та системах колективного керування [13].

Рекурентні нейронні мережі, зокрема LSTM та GRU, є більш ефективними при роботі з послідовними даними, оскільки враховують залежності у часі. Вони добре показують себе у прогнозуванні руху, моделюванні поведінки агента або оцінці динаміки навколишнього середовища. Їхня ключова перевага полягає у здатності працювати з шумними, неповними або нестабільними часовими даними, що є характерним для роботи реальних робототехнічних систем, однак ці архітектури мають обмеження у масштабованості та не забезпечують природної взаємодії між кількома агентами.

Наступними є графові нейронні мережі (GNN), що є одним із найбільш перспективних інструментів для систем групового керування. Вони дозволяють моделювати систему мобільних роботів як граф, де вузли відповідають окремим агентам, а ребра – взаємодіям між ними (наприклад, обмін інформацією або просторовій близькості). Саме GNN забезпечують природну масштабованість, адаптивність та здатність до децентралізованого прийняття рішень, що робить їх оптимальними у мультиагентних системах. Їхнім основним недоліком є висока обчислювальна складність, особливо при великій кількості роботів [14].

Окрему групу складають нейромережеві архітектури, орієнтовані на навчання з підкріпленням (Reinforcement Learning, RL), такі як DQN, PPO, A3C, SAC. Вони забезпечують здатність робота або групи роботів самостійно навчатися прийняттю оптимальних дій на основі винагороди. У контексті невизначених середовищ саме RL-політики демонструють найкращу адаптивність та здатність до прийняття рішень у складних ситуаціях. Поєднання RL із GNN дає змогу реалізувати мультиагентне навчання (MARL), у якому кожен робот навчається не лише враховувати навколишнє середовище, а й координувати свої дії з іншими агентами.

Крайнім досить потужним, хоча й обчислювально дорогим інструментом є Transformer-архітектури, що базуються на механізмі самоуваги, що дозволяють моделювати складні залежності між елементами вхідних даних та можуть застосовуватися для планування маршрутів, прогнозування поведінки перешкод або аналізу складних карт місцевості, через значні вимоги до обчислювальних ресурсів їх використання у реальних мобільних роботах часто є обмеженим.

Порівняння всіх розглянутих архітектур за основними критеріями – здатністю працювати в умовах невизначеності, масштабованістю, ефективністю обробки просторово-часових даних та придатністю для мультиагентних систем – дозволяє зробити висновок, що найбільш ефективним підходом є комбінація графових нейронних мереж та методів навчання з підкріпленням. GNN забезпечують природну модель взаємодії між роботами, тоді як RL дозволяє формувати політики оптимальних дій навіть у мінливому та непередбачуваному середовищі [15].

1.5 Постановка задач досліджень

Проблематика керування групою мобільних роботів в умовах невизначеності пов'язана з необхідністю одночасного врахування декількох чинників: неповноти інформації про навколишнє середовище, динамічної зміни перешкод, можливих похибок сенсорів, взаємного впливу роботів один на одного та обмеженості обчислювальних ресурсів кожного агента. Традиційні методи планування маршруту часто не забезпечують достатньої гнучкості та адаптивності для роботи в таких умовах, особливо у випадку колаборативної взаємодії кількох мобільних роботів. Метою роботи є підвищення ефективності побудови маршруту переміщення колаборативних роботів при груповому керуванні. Об'єктом дослідження є процес побудови маршруту для колаборативних роботів в умовах групового керування. Предметом дослідження є моделі, методи та програмне забезпечення для численного моделювання маршруту переміщення групи колаборативних роботів. Враховуючи проведений аналіз, у рамках кваліфікаційної роботи необхідно вирішити такі задачі дослідження:

- сформулювати вимоги до математичної моделі групи мобільних роботів, що повинна враховувати кінематичні та динамічні параметри роботів, обмеження на їхнє переміщення, можливість децентралізованої взаємодії та адаптації до змін середовища;

- розробити математичну модель опису мобільного робота, що включатиме його конструктивні та функціональні особливості, тип рушіїв, параметри сенсорної системи та можливі джерела похибок у вимірюваннях;
- створити математичне представлення навколишнього середовища, яке може містити статичні та динамічні перешкоди, зони ризику, області з низькою точністю сенсорних даних та маршрути інших роботів;
- сформулювати задачу прийняття рішень групою мобільних роботів у вигляді моделі мультиагентної взаємодії, в якій кожен агент діє на основі локальної інформації та навченої політики поведінки;
- розробити нейронну мережу або її архітектуру для реалізації процесу прийняття рішень, що забезпечує прогнозування оптимальної траєкторії або вибір дії в умовах невизначеності та мінливого середовища;
- сформувати метод групового керування мобільними роботами, який поєднуватиме математичні моделі та нейронні мережі, підтримуючи децентралізованість, масштабованість та можливість колаборативної взаємодії;
- обґрунтувати вибір програмних засобів, мови програмування та середовища моделювання, придатних для реалізації алгоритмів планування маршруту та навчання нейронної мережі;
- розробити алгоритм функціонування програмного модуля, що забезпечуватиме симуляцію руху роботів, обмін інформацією, обчислення маршрутів та оцінку рішень;
- провести моделювання роботи групи роботів, проаналізувати ефективність побудованих маршрутів, стійкість до невизначеності та якість взаємодії між агентами.

2 РОЗРОБКА МАТЕМАТИЧНОЇ МОДЕЛІ ФОРМАЛІЗАЦІЇ МОБІЛЬНОГО РОБОТА

2.1 Формування вимог до математичної моделі

Обґрунтування необхідності формування чітких вимог до математичної моделі мобільного робота та представлення навколишнього середовища при розробці програмного модуля моделювання керування групою мобільних роботів на базі нейронної мережі полягає у забезпеченні достовірності, відтворюваності та точності результатів моделювання. Математична модель повинна точно описувати кінематичні та динамічні параметри кожного робота, такі як маса, розміри, розподіл маси, швидкість, прискорення, енергоспоживання та обмеження рухливості, що дозволить нейронній мережі враховувати реальні фізичні обмеження та забезпечити коректне планування траєкторій та розв'язання задачі уникнення зіткнень. Опис навколишнього середовища є критично важливим, оскільки будь-які неточності або спрощення можуть призвести до формування неадекватної стратегії колективної взаємодії роботів. Структуроване представлення простору, включаючи перешкоди, динамічні об'єкти та змінні умови, дозволяє нейронній мережі розвивати здатність до адаптивного реагування на непередбачувані зміни та підтримувати ефективну координацію між агентами. Формування вимог також передбачає стандартизацію вхідних і вихідних параметрів, що спрощує інтеграцію моделі у більші системи керування та полегшує багатократне тестування сценаріїв. Без чітко визначених вимог існує ризик появи помилкових рішень, що негативно впливають на загальну безпеку та ефективність групи мобільних роботів. Наявність формалізованих параметрів дозволяє точно оцінювати якість рішень нейронної мережі, проводити об'єктивні порівняння альтернативних стратегій і визначати потенційні точки оптимізації. Це створює основу для масштабування системи, підвищення її надійності та здатності виконувати завдання в реальних умовах. Відповідно до вище перерахованого та керуючись параметрами, що

задані в технічному завданні (ТЗ) до апаратної складової мобільного робота, пропонуються наступні вимоги до математичної моделі мобільного робота, які представлено в таблиці 2.1.

Таблиця 2.1 – Формування вимог до математичної моделі мобільного робота

Параметр / підсистема	Вимоги до моделі	Призначення
Платформа 4WD	Модель кінематики та динаміки з урахуванням незалежного приводу кожного колеса; моделювання розподілу ваги та ковзання	Забезпечує точну траєкторію руху та стабільність
Двигуни DC 6 В	Модель електромеханічної динаміки двигунів, включаючи інерцію, коефіцієнти передавання та обмеження моменту	Враховується затримка та нелінійності
Драйвер L298N	Модель затримки, обмеження струму та втрат напруги на драйвері	Потрібно для адекватного керування швидкістю та моментом
Система керування (Raspberry Pi 4)	Модель обчислювальної затримки, обробки сигналів сенсорів та виконання команд	Важливо для оцінки реакції системи
Камера PiCam	Модель поля зору, точності розпізнавання об'єктів та частоти кадрів	Для моделювання зору та локалізації
Ультразвукові датчики HC-04	Модель дальності, кутового огляду, похибок вимірювань та затримок	Забезпечує обхід перешкод і побудову карти
Маніпулятор (3-ланковий)	Модель кінематики та динаміки кожного сегмента, обмеження по кутах і швидкостях	Необхідно для точного позиціонування і рухів
Серводвигуни 90g	Модель передавання зусилля, точності позиціонування, динаміки приводу	Забезпечує надійне керування маніпулятором
Захватний пристрій	Модель сили захвату, контактної взаємодії та обмежень по деформації	Для моделювання маніпуляцій з об'єктами

Формування детальних вимог до математичної моделі мобільного робота на базі таблиці 2.1, дає можливість створити адекватний та структурований математичний опис, що дозволяє точно відобразити фізичні та функціональні властивості всієї конструкції під час розробки системи моделювання керування групою мобільних роботів. Наявність опису платформи 4WD із урахуванням кінематики та динаміки забезпечує достовірне моделювання переміщення та стійкості на різних типах поверхні, що критично важливо для групового маневрування та уникнення зіткнень. Введення моделей електромеханічної поведінки двигунів і драйвера дозволяє врахувати обмеження по струму, моменту та затримкам, що формує більш реалістичну картину розподілу потужності між роботами у групі. Опис системи керування Raspberry Pi 4 та сенсорних модулів (камери й ультразвукових датчиків) формує основу для алгоритмів колективної навігації та прийняття рішень у режимі реального часу. Модель маніпулятора та захватного пристрою дозволяє деталізувати сценарії колективних маніпуляцій об'єктами, де кожен робот може виконувати свою частину завдання без конфліктів і перевантажень. Включення моделей середовища із перешкодами та змінними об'єктами критично необхідне для тренування нейронної мережі в умовах, наближених до реальних, забезпечуючи адаптивність і гнучкість групового керування. Завдяки таким вимогам ми отримуємо математичний фундамент, який мінімізує ризики некоректної поведінки під час колективної роботи, підвищує ефективність планування траєкторій і дає змогу будувати складні багаточарові стратегії взаємодії між роботами, що у підсумку робить модель практично застосовною для реальних сценаріїв.

2.2 Розробка математичної моделі опису параметрів мобільного робота

Для ефективної розробки програмного модуля моделювання керування групою мобільних роботів на базі нейронної мережі необхідно створити точну та комплексну математичну модель, яка детально описує параметри кожного мобільного робота. Така модель дозволяє врахувати кінематичні та динамічні

особливості руху, роботу силових установок, функціонування сенсорних систем та маніпуляторів, а також інтеграцію засобів сприйняття навколишнього середовища. Завдяки математичному опису забезпечується можливість прогнозування поведінки роботів у реальних умовах, моделювання групової взаємодії та навчання нейронної мережі з урахуванням фізичних обмежень. Це створює фундамент для розробки інтелектуальних алгоритмів колективної навігації, координації та виконання складних завдань, що є ключовим етапом у побудові ефективної системи моделювання та керування [3].

Представимо кінематичну модель платформи 4WD у вигляді наступного виразу:

$$\begin{cases} \dot{x} = v \cos \theta \\ \dot{y} = v \sin \theta, \\ \dot{\theta} = w \end{cases} \quad (2.1)$$

де x – координата центра мас платформи за віссю X у глобальній (інерціальній) системі координат, визначає положення робота по горизонталі на площині;

y – координата центра мас платформи за віссю Y у глобальній системі координат, вказує положення робота по вертикалі на площині;

θ – кут орієнтації робота відносно осі X глобальної системи координат, визначає напрямок «голови» або передньої частини платформи;

$\dot{x}$ – швидкість зміни координати x у часі, вказує на проєкцію лінійної швидкості v на вісь X ;

$\dot{y}$ – швидкість зміни координати y у часі, відповідає проєкції лінійної швидкості v на вісь Y ;

v – лінійна швидкість центра мас платформи (уздовж напрямку її орієнтації). Це результуюча швидкість руху робота вперед чи назад;

w – кутова швидкість обертання робота навколо вертикальної осі (Z -осі), вказує швидкість зміни кута орієнтації θ ;

$\cos \theta, \sin \theta$ – коефіцієнти перетворення лінійної швидкості v у проєкції на глобальні координати X і Y відповідно.

Запропонована модель (2.1) дає можливість описати траєкторію переміщення центра мас мобільного робота на площині, враховуючи його напрямок руху й можливість обертання. Ця модель дає змогу прогнозувати положення робота в будь-який момент часу та формує основу для побудови алгоритмів планування руху та навігації в групових сценаріях.

Розробимо динамічну модель платформи 4WD у вигляді наступного виразу:

$$\begin{aligned} M\dot{v} &= F - F_{tp} \\ J\dot{\omega} &= M_z \end{aligned} \quad (2.2)$$

де M – маса мобільного робота, визначає інерційні властивості при поступальному русі, впливає на здатність робота розганятись або гальмувати;

$\dot{v}$ – лінійне прискорення центра мас робота (перша похідна лінійної швидкості v за часом), показує, наскільки швидко змінюється швидкість поступального руху;

F – сумарне тягове зусилля, створюване колесами робота, це основна рушійна сила, що забезпечує поступальний рух уперед або назад;

F_{tp} – сила опору руху (сила тертя кочення, ковзання, аеродинамічного та інших типів опорів), зменшує ефективність руху та визначає необхідність компенсації для підтримання стабільної швидкості;

J – момент інерції мобільного робота відносно вертикальної осі (ось обертання), характеризує інерційні властивості при обертанні, впливає на реакцію робота на крутний момент;

$\dot{\omega}$ – кутове прискорення (перша похідна кутової швидкості ω за часом), показує, наскільки швидко змінюється швидкість обертання робота;

M_z – сумарний обертальний момент, прикладений до робота (наприклад, різниця тягових сил на колесах зліва і справа), викликає зміну кута орієнтації робота.

Призначення моделей (2.2) полягає в описі реакції мобільного робота на зовнішні та внутрішні сили та моменти, тобто динаміки розгону, гальмування та

обертання. Така модель потрібна для точного розрахунку силових впливів, прогнозування руху при керуванні групою роботів і навчання нейронної мережі адекватно оцінювати реальні обмеження системи.

Електромеханічну модель двигуна DC 6 В можна представити у вигляді наступної моделі:

$$L \frac{di}{dt} + Ri = U - k_e \omega_m; \quad (2.3)$$

$$J_m \frac{d\omega_m}{dt} = k_m i - T_{load}; \quad (2.4)$$

де L – індуктивність обмотки двигуна [Гн], характеризує здатність обмотки протидіяти зміні струму, тобто впливає на динаміку наростання струму при зміні напруги;

$\frac{di}{dt}$ – швидкість зміни струму в обмотці за часом [А/с], показує, наскільки швидко струм реагує на прикладену напругу;

R – електричний опір обмотки двигуна [Ом], визначає втрати енергії на нагрівання та впливає на стабільність роботи двигуна;

i – струм, що протікає через обмотку двигуна [А], прямо пов'язаний зі створенням обертового моменту на валу;

k_e – коефіцієнт електрорушійної сили [В·с/рад], показує, яка напруга генерується обмоткою при обертанні ротора (зворотний вплив швидкості обертання на електричну частину);

ω_m – кутова швидкість ротора двигуна [рад/с], визначає швидкість обертання валу двигуна, безпосередньо впливає на швидкість колеса;

U – прикладена напруга живлення двигуна [В], є основним керуючим сигналом для регулювання швидкості обертання;

J_m – момент інерції ротора двигуна [кг·м²], характеризує здатність ротора протидіяти зміні кутової швидкості, тобто впливає на плавність розгону та гальмування;

$\frac{dw_m}{dt}$ – кутове прискорення ротора [рад/с²], показує, як швидко змінюється швидкість обертання валу;

b – коефіцієнт в'язкого тертя в двигуні [Н·м·с/рад], описує втрати на внутрішнє тертя під час обертання;

k_m – коефіцієнт моменту [Н·м/А], визначає, який обертальний момент створюється на валу двигуна при даному струмі.

Призначення моделі (2.3) та (2.4), полягає в точному описі взаємозв'язку між електричними параметрами двигуна та механічною реакцією (обертанням валу). Вона дозволяє моделювати, як двигун реагує на зміну напруги, враховуючи інерційні властивості та електричні обмеження. Такий опис критично важливий для розробки програмного модуля моделювання керування, щоб нейронна мережа могла навчитися оптимально управляти моментами та швидкостями в реальних умовах, запобігаючи перенавантаженню або нестабільним режимам.

Модель представлення драйвера L298N, можна представити у вигляді обмеження напруги та струму:

$$U_{out} = U_s - \Delta U, \quad (2.5)$$

де U_{out} – вихідна напруга на клеммах двигуна [В], це фактична напруга, яка прикладається безпосередньо до двигуна після проходження через драйвер та визначає, яку швидкість і момент зможе розвинути двигун;

U_s – вхідна (живильна) напруга драйвера [В], це напруга, яка подається на драйвер від акумулятора або джерела живлення, та задає максимальну межу доступної напруги для двигуна;

ΔU – втрати напруги на драйвері [В], цей параметр враховує внутрішні падіння напруги на транзисторах драйвера та паразитні опори всередині мікросхеми, та залежить від струму навантаження (i): чим вищий струм через двигун, тим більшими будуть ці втрати, з умовами наступного обмеження по струму:

$$i \leq i_{max}. \quad (2.6)$$

Модель (2.5) дає можливість відображення реальної ефективності драйвера та його обмежень при передачі напруги до двигуна. Навіть якщо на вході подається висока напруга U_s , на виході U_{out} завжди буде менше через неминучі втрати ΔU . Ця модель дає можливість правильно моделювати роботу двигуна під навантаженням, прогнозувати зниження швидкості, нагрів і зменшення ефективності. Для розробки програмного модуля моделювання керування групою роботів цей параметр критично важливий, оскільки дозволяє нейронній мережі враховувати реальні обмеження електроніки при плануванні руху й розподілі моментів між колесами.

Система керування мобільним роботом побудована на базі Raspberry Pi 4, розробимо математичну модель у вигляді моделі затримки управління:

$$\tau_d = \tau_{comp} + \tau_{sens} + \tau_{execu}, \quad (2.7)$$

де τ_d – загальна затримка системи керування [с], це сумарний час від моменту отримання інформації від сенсорів до фактичного виконання команди приводами та визначає швидкість реакції робота на зовнішні події;

τ_{comp} – час обчислення [с], це час, потрібний процесору Raspberry Pi для обробки вхідних даних, виконання алгоритмів планування руху, прийняття рішень або обробки нейронної мережі та залежить від складності програмних алгоритмів та потужності апаратного забезпечення;

τ_{sens} – час збору та обробки сенсорної інформації [с], включає затримку на отримання сигналів з камери, ультразвукових датчиків або інших сенсорів, а також попередню фільтрацію або обробку даних;

τ_{execu} – час виконання команд [с], це час, необхідний для передачі команд з процесора на драйвери двигунів або сервоприводи та безпосередній запуск виконавчих механізмів (наприклад, зміни швидкості обертання коліс або положення маніпулятора).

Призначення моделі (2.7) полягає в описі затримок, які виникають у всьому контурі керування мобільним роботом. Вона дозволяє врахувати інерцію обчислювального процесу, обмеження сенсорної системи й фізичну інертність виконавчих механізмів. Це особливо важливо для розробки програмного модуля моделювання керування групою роботів на базі нейронної мережі, оскільки правильне врахування затримок забезпечує адекватне прогнозування дій робота в реальному часі, підвищує точність траєкторій і стабільність групової взаємодії.

Модель PiCam камери представимо у вигляді опису поля зору:

$$FOV = \left[-\frac{\alpha}{2}, \frac{\alpha}{2} \right], \quad (2.8)$$

де FOV – (Field of View) – поле зору камери (градуси або радіани). Це кутовий діапазон, в межах якого камера здатна «бачити» об'єкти перед собою. Визначає ширину сцени, яку робот може одночасно сприймати;

α – повний кут огляду камери (градуси або радіани), це основний параметр оптики, який задає максимальну ширину поля зору. Чим більше значення α , тим ширший простір може охоплювати камера, але тим менше деталізація далеких об'єктів;

$\left[-\frac{\alpha}{2}, \frac{\alpha}{2} \right]$ – діапазон кутів відносно центральної осі оптики камери, цей інтервал показує, що камера може бачити об'єкти ліворуч і праворуч від центральної лінії огляду на половину кута α .

Моделі (2.8) дає можливість провести опис геометричних обмежень сприйняття камери, тобто зони, в межах якої робот може отримувати візуальні дані для навігації, виявлення об'єктів чи побудови карти. Для розробки програмного модуля моделювання керування групою мобільних роботів на базі нейронної мережі цей параметр критично важливий, оскільки він визначає інформаційне поле робота: які об'єкти будуть доступні для аналізу в кожен момент часу, як планувати траєкторію, уникати перешкод і координуватися з іншими агентами.

Тоді модель розпізнавання можна представити наступним чином:

$$P_d = f(L, d, Q), \quad (2.9)$$

де P_d – ймовірність детекції об'єкта, це вихідний показник моделі, який характеризує, з якою ймовірністю камера або система комп'ютерного зору розпізнає об'єкт у кадрі за певних умов. Значення P_d лежить у межах від 0 (об'єкт точно не буде розпізнаний) до 1 (об'єкт гарантовано розпізнаний);

L – рівень освітленості сцени (наприклад, у люксах), визначає, наскільки чітко та контрастно камера сприймає об'єкти. За низького рівня освітлення P_d суттєво зменшується через шум та втрату деталей;

d – відстань до об'єкта (метри або сантиметри), впливає на розмір об'єкта в кадрі та якість його відображення. Зі збільшенням d об'єкт займає менше пікселів, що ускладнює впевнене розпізнавання та зменшує P_d ;

Q – якість оптики або загальна якість зображення (може характеризуватись, наприклад, роздільною здатністю камери, співвідношенням сигнал/шум, якістю обробки). Вища якість дає змогу чіткіше бачити контури й деталі об'єкта, збільшуючи P_d ;

Призначення моделі (2.9) полягає в тому, щоб описати вплив фізичних та оптичних факторів на здатність камери розпізнавати об'єкти. Такий опис дозволяє моделювати реальні сценарії, коли робот може або не може вчасно «побачити» перешкоду чи ціль, що критично для планування безпечної траєкторії та колективної координації. Для розробки програмного модуля моделювання керування групою мобільних роботів на базі нейронної мережі ця модель потрібна, щоб навчити систему враховувати обмеження зору й адаптивно приймати рішення у складних середовищах.

Модель вимірювання відстані на базі ультразвукових датчиків НС-04 опишемо наступним чином:

$$d_m = d + \epsilon_d, \quad (2.10)$$

де d_m – виміряна відстань (метри або сантиметри), це фактичне значення, яке отримує робот від ультразвукового датчика. Використовується для побудови карти навколишнього середовища, виявлення перешкод і планування руху;

d – реальна відстань до об'єкта (метри або сантиметри), це фізично точна відстань між сенсором і найближчим об'єктом у напрямку випромінювання ультразвукового імпульсу;

ϵ_d – похибка вимірювання (метри або сантиметри), це випадкова або систематична помилка, яка виникає внаслідок шуму, відбиття хвиль, обмеженої роздільної здатності сенсора або впливу середовища (наприклад, температура, вологість, кут падіння хвилі). Часто моделюється як випадкова величина, наприклад, нормальний розподіл $N(0, \sigma_d^2)$.

Модель (2.10) полягає у відображенні реалістичної роботи ультразвукового сенсора, з урахуванням неминучих похибок і шумів. Це дозволяє точно змодельовати, як робот сприймає відстані до перешкод або об'єктів, і врахувати ризик помилкових рішень під час навігації. Для розробки програмного модуля моделювання керування групою мобільних роботів на базі нейронної мережі така модель критично важлива, оскільки навчає систему працювати з неточними даними, компенсувати похибки сенсорів і приймати більш стійкі рішення у динамічних та невизначених середовищах.

Математичну модель 3-ланкового маніпулятора представимо в загальному виді у вигляді кінематичної моделі:

$$\begin{cases} x = l_1 \cos \theta + l_2 \cos(\theta_1 + \theta_2) + l_3 \cos(\theta_1 + \theta_2 + \theta_3) \\ y = l_1 \sin \theta + l_2 \sin(\theta_1 + \theta_2) + l_3 \sin(\theta_1 + \theta_2 + \theta_3) \end{cases} \quad (2.11)$$

де x – координата кінцевого ефектора (захвата) маніпулятора по горизонтальній осі, визначає положення робочого інструмента маніпулятора у площині;

y – координата кінцевого ефектора по вертикальній осі. Разом з x описує точку досяжності або позицію робочої точки в робочому просторі;

l_1 – довжина першої (базової) ланки маніпулятора, визначає досяжність маніпулятора і впливає на загальну конфігурацію при розгортанні;

l_2 – довжина другої ланки, додає ступінь свободи та розширює робочий простір маніпулятора, впливаючи на точність позиціонування;

l_3 – довжина третьої ланки (кінцевої), забезпечує гнучкість при захваті об'єктів у різних точках простору, впливає на можливість обгинання перешкод;

θ_1 – кут повороту першої ланки відносно бази (рад або градуси), задає основну орієнтацію всієї системи та визначає напрямок подальшого розгортання;

θ_2 – кут між першою і другою ланкою, визначає форму середньої частини маніпулятора, впливаючи на розташування кінцевого ефектора;

θ_3 – кут між другою і третьою ланкою, дозволяє точно орієнтувати захват відносно об'єкта, важливий для виконання маніпуляційних задач.

Призначення моделі (2.11) полягає в описі положення кінцевої точки маніпулятора у двовимірному просторі як функції довжин ланок і кутів суглобів. Така модель дозволяє прогнозувати траєкторії руху кінцевого ефектора, розраховувати зони досяжності та забезпечувати точне управління під час захоплення або маніпуляцій. Для програмного модуля моделювання керування групою роботів на базі нейронної мережі ця модель критично необхідна, оскільки навчає систему враховувати кінематичні обмеження при плануванні колективних маніпуляцій, взаємодії з об'єктами та уникненні конфліктів при груповій роботі.

Модель серводвигунів SG90, які використовуються як виконавчі органи в 3-ланковому маніпуляторі можна представити з точки зору моделі позиціонування:

$$\theta(t) = \theta_0 + (\theta_f - \theta_0)(1 - e^{-t/\tau_s}), \quad (2.12)$$

де $\theta(t)$ – кут положення серводвигуна у момент часу t (градуси або рад), це фактичний кут, на який повернено вал сервоприводу в заданий момент часу. Визначає реальну позицію ланки маніпулятора;

θ_0 – початковий кут положення серводвигуна (градуси або рад), це кут, з якого сервопривід починає рух при виконанні команди;

θ_f – фінальний (цільовий) кут, до якого має дійти серводвигун (градуси або рад), визначає кінцеву позицію ланки після виконання руху;

t – час (с) від початку руху. Показує, на якому етапі переходу від θ_0 до θ_f знаходиться сервопривід;

τ_s – постійна часу серводвигуна (с), характеризує швидкість реакції серводвигуна на команду. Менше значення τ_s означає швидший відгук і швидше досягнення цільового кута;

e^{-t/τ_s} – експоненційний член, що описує плавність динаміки переходу. Враховує інерційні та електромеханічні властивості серводвигуна, забезпечуючи поступове наближення до цільового кута без різких стрибків.

Модель (2.12) дає змогу описати динамічний процес позиціонування серводвигуна під час виконання руху. Вона дозволяє прогнозувати траєкторію кута в часі та враховувати затримки, інерцію й плавність руху. Для програмного модуля моделювання керування групою мобільних роботів на базі нейронної мережі ця модель важлива, оскільки дає змогу точно імітувати реальну поведінку маніпулятора при виконанні складних маніпуляцій, забезпечуючи узгодженість і координацію рухів між різними роботами в групі.

Модель захватного пристрою, представимо через силу захвату:

$$F_g = k_g(\Delta x), \quad (2.13)$$

де F_g – сила захвату [Н], це результуюча сила, з якою захватний механізм притискає або утримує об'єкт. Визначає здатність надійно утримувати предмети різних мас і розмірів під час маніпуляцій;

k_g – жорсткість механізму захвату [Н/м], це коефіцієнт пружності, що характеризує, наскільки сильно захват чинить протидію деформації. Від цього параметра залежить чутливість і сила стискання: чим більший k_g , тим швидше нарастає сила при малій деформації;

Δx – відхилення або деформація захватного механізму [м], це різниця між початковою (нейтральною) відстанню між пальцями захвату та фактичною відстанню під час контакту з об'єктом. Визначає, наскільки сильно захват стиснув об'єкт.

Призначення моделі (2.13) полягає у відображенні процесу формування сили утримання об'єкта залежно від механічної деформації захвату. Модель дає змогу оцінити, чи буде об'єкт надійно утримуватись при заданих характеристиках жорсткості й геометрії. Для програмного модуля моделювання керування групою мобільних роботів на базі нейронної мережі ця модель є критично важливою, оскільки дозволяє врахувати реальні фізичні обмеження під час захвату об'єктів, запобігти їх вислизанню або пошкодженню, а також забезпечити координацію колективних маніпуляцій кількома роботами [3].

Переваги розроблених математичних моделей із точки зору даного дослідження полягають у можливості комплексного та реалістичного опису фізичних і функціональних властивостей кожної підсистеми робота. Використання детальної кінематичної та динамічної моделі руху платформи дозволяє точно прогнозувати траєкторії та швидкість пересування, що є критично важливим для планування групових маневрів і уникнення зіткнень. Електромеханічна модель двигунів і моделі драйверів дають змогу врахувати реальні обмеження привідної системи, втрати напруги та інерційні ефекти, що забезпечує більш точну симуляцію дій в умовах змінних навантажень. Моделі сенсорів, зокрема камери PiCam і ультразвукових датчиків HC-04, дозволяють моделювати обмеження зору та точності вимірювань, навчаючи нейронну мережу працювати в умовах невизначеності та шуму. Включення моделі маніпулятора та захватного пристрою забезпечує реалістичне планування маніпуляційних дій, дозволяючи групі роботів координовано взаємодіяти з об'єктами. Окремо важливими є моделі затримок системи керування на базі Raspberry Pi, які враховують обчислювальні, сенсорні та виконавчі затримки, наближаючи модуль моделювання до умов реального часу. Використання цих моделей створює потужну основу для навчання та перевірки нейронної мережі, дозволяючи розробити гнучкі та стійкі алгоритми колективного керування,

підвищити безпечність, адаптивність і ефективність групових дій мобільних роботів.

2.3 Розробка математичного представлення навколишнього середовища

В умовах колективної роботи мобільних агентів важливо забезпечити точне врахування характеристик рухомих перешкод, властивостей сенсорного сприйняття та можливих обмежень огляду, що визначають успішність виконання групових завдань. Формування математичних моделей, які описують середовище, дозволяє реалізувати більш гнучкі алгоритми планування руху, уникнення зіткнень та маніпуляцій із об'єктами. Такий підхід створює підґрунтя для навчання нейронних мереж, що повинні враховувати змінні умови реального світу та обмеження апаратної платформи. Математичне представлення середовища дозволяє прогнозувати сценарії поведінки як окремого робота, так і всієї групи, забезпечуючи підвищення ефективності, безпечності та координації між агентами. Завдяки цьому можна максимально наблизити модельні експерименти до реальних ситуацій і значно полегшити подальше впровадження розроблених алгоритмів у фізичні прототипи роботів.

В рамках даних досліджень, представимо навколишнє середовище в момент часу t :

$$\mathcal{E}(t) = \{O_i(t), \forall i = 1, \dots, N\}, \quad (2.14)$$

де $\mathcal{E}(t)$ – навколишнє середовище в момент часу t , це узагальнена множина всіх об'єктів, які присутні в робочій зоні мобільного робота. Використовується для математичного опису просторового розташування та поведінки об'єктів, з якими робот може взаємодіяти або які слугують перешкодами;

$O_i(t)$ – i -й об'єкт середовища в момент часу t , це абстрактне подання кожного окремого об'єкта, що входить до складу середовища. Кожен об'єкт описується своїми параметрами положення, розміру, швидкості руху тощо.

$i = 1, \dots, N$ – індекс об'єкта. Вказує номер об'єкта в множині всіх об'єктів середовища, де N – загальна кількість об'єктів, присутніх у середовищі в поточний момент часу.

Проведемо розробку моделі формалізацій i -го об'єкта $O_i(t)$ в середовищі в момент часу t . Загальна модель $O_i(t)$ пропонується наступною:

$$O_i(t) = (x_i(t), y_i(t), v_{xi}(t), v_{yi}(t), S_i, \phi_i(t), T_i), \quad (2.15)$$

де $x_i(t), y_i(t)$ – координати центра мас об'єкта i у глобальній системі координат [м], визначають поточне положення об'єкта у площині робочої зони, необхідні для оцінки відстані до робота, планування траєкторії та ухилення від зіткнень;

$v_{xi}(t), v_{yi}(t)$ – компоненти лінійної швидкості об'єкта i у напрямках осей X та Y [м/с], описують динаміку руху об'єкта, тобто дає можливість прогнозування майбутніх положень і побудови динамічної карти середовища;

S_i – розмір або форма об'єкта, використовується для моделювання форми, оцінки зони можливого контакту, перевірки умов зіткнень або захоплення. Як приклад може бути представлено у вигляді наступної моделі:

$$S_i = \begin{cases} r_i, & \text{для круглих об'єктів} \\ (a_i, b_i), & \text{для прямокутних об'єктів} \end{cases}, \quad (2.16)$$

де r_i – радіус об'єкта; (a_i, b_i) – ширина та висота об'єкта.

$\phi_i(t)$ – орієнтація об'єкта i [рад або градуси], дає можливість описати кут положення або напрямку руху об'єкта, що важливо для об'єктів, які не є симетричними (наприклад, інші роботи чи піддон із вантажем);

T_i – тип i -го об'єкта, пропонується використовувати для класифікації об'єктів в робочій зоні мобільного робота:

$$T_i = \begin{cases} 1, \text{ціль (Target)} \\ 2, \text{статична перешкода} \\ 3, \text{рухома перешкода} \\ 4, \text{інший мобільний робот} \end{cases}, \quad (2.17)$$

Модель (2.17) дозволяє роботу відрізняти об'єкти, планувати захват, об'їзд або координацію.

Враховуючи, що робоча зона мобільного робота є динамічною, то модель руху об'єкта можна представити наступним чином:

$$\begin{cases} \dot{x}_i(t) = v_{xi}(t) \\ \dot{y}_i(t) = v_{yi}(t) \\ \phi_i(t) = v_{\phi i}(t) \end{cases}, \quad (2.18)$$

де $\dot{x}_i(t) = v_{xi}(t)$ – рівняння для опису швидкості зміни координати x об'єкта i у часі, тобто показує, як швидко та в якому напрямку змінюється горизонтальне положення об'єкта та використовується для прогнозування майбутнього положення об'єкта на осі;

$\dot{y}_i(t) = v_{yi}(t)$ – рівняння для опису швидкості зміни координати y об'єкта i у часі, тобто відображає вертикальний компонент руху об'єкта. Дає змогу визначати траєкторію по осі Y , що важливо для планування уникнення зіткнень або зближення;

$\phi_i(t) = v_{\phi i}(t)$ – рівняння для опису зміни орієнтації (кута) об'єкта i у часі, та показує, як швидко змінюється напрямок або кутова орієнтація об'єкта. Даний параметр є особливо важливим для об'єктів, які можуть обертатись або мають направлену форму (наприклад, інші мобільні роботи чи платформи);

$v_{\phi i}(t)$ – кутова швидкість об'єкта i , тобто швидкість зміни кута ϕi [рад/с].

Запропонована модель (2.18) дозволяє формально описати динамічну поведінку об'єктів у середовищі: їхнє лінійне переміщення по координатах та обертання. Завдяки цьому робот може прогнозувати позицію та орієнтацію кожного об'єкта в наступні моменти часу, що критично важливо для планування

руху, навігації, взаємодії маніпулятора, розпізнавання об'єктів камерою та уникнення зіткнень у колективних сценаріях.

Умова досяжності об'єкта для маніпулятора, представлена нижче:

$$\sqrt{(x_e(t) - x_i(t))^2 + (y_e(t) - y_i(t))^2} \leq \delta, \quad (2.19)$$

де $(x_e(t))$ – координата кінцевого ефектора маніпулятора по осі X у момент часу t [м], та вказує положення кінцевої робочої точки маніпулятора (захватного пристрою), яка безпосередньо взаємодіє з об'єктом;

$y_e(t)$ – координата кінцевого ефектора маніпулятора по осі Y у момент часу t [м], та разом $(x_e(t))$ визначає точне положення захвату в площині робочої зони робота;

$x_i(t)$ та $y_i(t)$ – координата центра об'єкта i по осі X у момент часу t [м], та відповідно координата центра об'єкта i по осі Y у момент часу t ;

δ – допустима похибка [м], визначає радіус зони досяжності кінцевого ефектора навколо об'єкта, в межах якого захват вважається можливим і успішним. Параметр враховує механічні похибки, неточності позиціювання та габарити захватного пристрою, тобто чим менше значення δ , тим точніше повинен бути рух маніпулятора для виконання захвату.

Умова (2.19) формально визначає момент, коли кінцевий ефektor маніпулятора наблизився до об'єкта настільки близько, що можна здійснити фізичний контакт або захоплення. Вона використовується для автоматичного контролю завершення руху до цілі, активації захватного механізму або початку подальших маніпуляцій. В контексті групового керування мобільними роботами на базі нейронної мережі ця умова дозволяє алгоритму точно визначати момент взаємодії з об'єктом, що критично для координації дій між декількома роботами та уникнення конфліктів під час колективних маніпуляцій.

2.4 Висновки до 2 розділу

У розділі «Розробка математичної моделі формалізації мобільного робота» було послідовно сформульовано вимоги до математичної моделі, які визначають точність відображення кінематичних і динамічних характеристик платформи, маніпулятора, сенсорної системи та виконавчих механізмів. Розроблено математичну модель опису параметрів мобільного робота, яка враховує рух платформи 4WD, динаміку електромеханічних приводів, характеристики драйверів, роботу камери та ультразвукових датчиків, а також кінематику триланкового маніпулятора й захватного пристрою. Це дозволяє моделювати реальні фізичні обмеження, затримки керування та відхилення, що забезпечує більш достовірне відтворення поведінки робота під час виконання завдань. Окремо було розроблено математичне представлення динамічного навколишнього середовища, яке описує рухомі та статичні об'єкти, їхні параметри, траєкторії, швидкості та зони взаємодії з маніпулятором. Це представлення дає можливість врахувати складність та змінність умов, у яких діє робот, і закласти основу для навчання нейронної мережі працювати в умовах невизначеності. Всі розроблені моделі інтегровані в єдину структуру, яка дозволяє здійснювати симуляцію колективних дій групи мобільних роботів, оцінювати ефективність алгоритмів керування та оптимізувати стратегії руху і взаємодії. Таким чином, виконана робота створює надійне математичне підґрунтя для побудови програмного модуля моделювання, спрямованого на підвищення адаптивності, безпечності та координації багатоагентних систем.

3 РОЗРОБКА МЕТОДА КЕРУВАННЯ ГРУПОЮ МОБІЛЬНИХ РОБОТІВ

3.1 Розробка математичної моделі прийняття рішень групою мобільних роботів на базі нейронної мережі

У сучасних системах колективної робототехніки ключову роль відіграє здатність кожного мобільного робота приймати швидкі та обґрунтовані рішення, адаптуючись до змін середовища та координуючи свої дії з іншими агентами. Це особливо актуально для сценаріїв, де необхідно досягти високого рівня автономності та безпечності при одночасному виконанні складних завдань у групі. Використання нейронних мереж як ядра прийняття рішень відкриває нові можливості для гнучкого планування траєкторій, врахування взаємодії між роботами та ефективного обходу перешкод. Розробка математичної моделі прийняття рішень на основі навчання дозволяє враховувати нелінійні залежності між параметрами середовища, станом роботів та динамікою цілей, що істотно підвищує адаптивність системи. Застосування такого підходу забезпечує здатність групи мобільних роботів колективно оцінювати ситуацію, обирати оптимальні маршрути та своєчасно реагувати на появу нових об'єктів чи загроз. В основі цього процесу лежить формалізація стану кожного мобільного робота (S_j), яка поєднує інформацію про власні параметри руху, сенсорні дані та цільові орієнтири, створюючи математичний фундамент для інтеграції в архітектуру нейронної мережі, яка представлена в наступному вигляді:

$$S_j(t) = (x_j(t), y_j(t), \theta_j(t), v_j(t), w_j(t), d_{m,1}(t), \dots, d_{m,M}(t), I_j(t)), \quad (3.1)$$

де $x_j(t)$ – координата робота j по осі X у момент часу t [м], описує поточне горизонтальне положення робота в робочій зоні, необхідне для планування руху та взаємодії з іншими агентами або цілями;

$y_j(t)$ – координата робота j по осі Y у момент часу t [м], вказує вертикальне положення робота, що дозволяє визначати його місце на площині середовища;

$\theta_j(t)$ – орієнтація робота j (рад або градуси), відображає напрямок, у якому "дивиться" або рухається робот та використовується для обчислення напрямку руху та планування поворотів;

$v_j(t)$ – лінійна швидкість робота j у момент часу t [м/с], описує швидкість поступального руху робота, необхідна для обчислення подальших позицій і оцінки енергоспоживання;

$w_j(t)$ – кутова швидкість робота j у момент часу t [рад/с], визначає, наскільки швидко робот змінює орієнтацію, тобто швидкість повороту та важливо для керування маневреністю;

$d_{m,1}(t), \dots, d_{m,M}(t)$ – множина відстаней, виміряних M сенсорами (наприклад, ультразвуковими датчиками) у момент часу t [м], відображають інформацію про відстань до найближчих перешкод або об'єктів у різних напрямках навколо робота та використовується для уникнення зіткнень і побудови локальної карти оточення;

$I_j(t)$ – інтегральна інформація від камер або інших візуальних сенсорів у момент часу t , може містити ймовірність виявлення цілі, класифікацію об'єктів або інші параметри розпізнавання. Використовується для уточнення поведінки робота щодо об'єктів або цілей.

Вектор стану формалізує повний поточний опис робота j , який включає просторове положення, динамічні параметри руху, інформацію про оточення та дані сенсорів. Це дозволяє нейронній мережі об'єктивно оцінювати ситуацію, прогнозувати подальші дії та обирати оптимальну траєкторію руху до цілі, забезпечуючи безпечну і скоординовану поведінку всієї групи мобільних агентів.

Проведемо формалізацію цілі, у вигляді наступного виразу:

$$G_j = (x_g, y_g), \quad (3.2)$$

де x_g – координата цілі по осі X [м], вказує горизонтальне положення точки призначення або робочої точки, до якої повинен наблизитися або яку повинен

досягти мобільний робот j , ця координата задає напрямок руху та є основою для планування траєкторії у площині;

y_g – координата цілі по осі Y [м], відображає вертикальне положення точки цілі у робочій зоні та разом із x_g формує точне місце, де робот має завершити завдання: наприклад, підібрати об'єкт, передати вантаж або виконати іншу маніпуляцію.

Вектор G_j формалізує просторові координати завдання робота j і слугує головним орієнтиром для обчислення поточної відстані, вибору напрямку руху та коригування траєкторії. Він є ключовим параметром у нейронній мережі для генерації керуючих дій, оскільки визначає кінцеву мету кожного індивідуального агента (робота) в групі.

Наступним кроком проведемо математичний опис структури нейронної мережи:

– вихід нейронної мережі, визначається як керуючі дії:

$$A_j(t) = (a_v(t), a_w(t)), \quad (3.3)$$

де $a_v(t)$ – керуюча дія по лінійній швидкості робота j у момент часу t [м/с² або м/с], визначає зміну або значення лінійної швидкості робота в поточний момент та використовується для контролю поступального руху вперед або назад, залежно від ситуації. Це дозволяє регулювати швидкість наближення до цілі, зупинку чи плавний об'їзд перешкод;

$a_w(t)$ – керуюча дія по кутовій швидкості робота j у момент часу t [рад/с² або рад/с], визначає зміну або значення кутової швидкості (обертання) робота та використовується для маневрування та орієнтації робота в напрямку цілі або при обході перешкод. Дозволяє адаптувати кутову динаміку, щоб точно слідувати обраній траєкторії чи уникати зіткнень.

Вектор $A_j(t)$ формалізує поточні команди керування, які видаються нейронною мережею для робота j . Він визначає, як змінити рух у кожен момент часу, інтегруючи інформацію про стан робота, ціль та оточення. Завдяки цим параметрам робот реалізує плавні й безпечні траєкторії, координує рухи з

іншими агентами та забезпечує ефективне досягнення задачі в динамічному середовищі.

– вхідний шар пропонується у вигляді стану $S_j(t)$ робота та координати цілі G_j :

$$Input = [x_j; y_j; \theta_j; v_j; d_j; d_{m,1}; \dots; d_{m,M}; I_j; x_g; y_g], \quad (3.4)$$

де x_j – координата робота j по осі X [м], вказує поточне горизонтальне положення робота в середовищі та використовується для оцінки відстані до цілі та розташування відносно перешкод;

y_j – координата робота j по осі Y [м], ;дозволяє визначати вертикальне положення робота;

θ_j – орієнтація робота j [рад або градуси], визначає напрямок руху робота, необхідний для планування поворотів і вибору траєкторії в напрямку цілі чи для об'їзду перешкод;

v_j – лінійна швидкість робота j [м/с], дозволяє врахувати поточну динаміку робота, що важливо для прогнозування можливостей гальмування, розгону та плавної зміни траєкторії;

d_j – поточна відстань до цілі [м], використовується для прямого оцінювання відстані до задачі або об'єкта, що треба досягти. Може бути розрахована як евклідова відстань:

$$d_j = \sqrt{(x_j - x_g)^2 + (y_j - y_g)^2}. \quad (3.5)$$

$d_{m,1}; \dots; d_{m,M}$ – відстані, виміряні M сенсорами (наприклад, ультразвуковими або лідаром) [м], дають інформацію про розташування перешкод навколо робота та допомагають мережі «бачити» локальне середовище й уникати зіткнень;

I_j – інтегральна або зведена інформація від візуального сенсора (наприклад, камери), може містити ймовірність виявлення цілі, ознаки об'єкта

або класифікаційні ознаки сцени та дає високорівневу інформацію для корекції руху або уточнення стратегії;

x_g – координата цілі по осі X [м], вказує горизонтальну позицію кінцевої цілі, потрібну для планування глобальної траєкторії;

y_g – координата цілі по осі Y [м], визначає вертикальну позицію кінцевої точки завдання.

Набір параметрів (3.4) формує комплексне уявлення про поточний стан робота, оточення та задачу. Вхідний шар є основою для того, щоб нейронна мережа навчилася приймати рішення, обирати керуючі дії, будувати траєкторію до цілі та уникати перешкод, забезпечуючи при цьому координацію в групі роботів.

Пропонується реалізувати чотири прихованих шарів:

- шар 1 (конкатенація сенсорних ознак): виявлення ознак від камер і ультразвукових датчиків, виділення ознак перешкод;

- шар 2 (просторове планування): генерація умовної карти навколишнього середовища та застосовується нелінійна активація ReLU;

- шар 3 (інтеграція траєкторії): об'єднання інформації про простір і ціль, оцінка потенційних траєкторій;

- шар 4 (рішення): обчислення остаточного набору керуючих команд $A_j(t)$.

Модель вихід нейронної мережі (3.2) приймає наступний вигляд:

$$A_j(t) = MLP(S_j(t), G_j), \quad (3.6)$$

де MLP – багатошаровий перцептрон, який регресує два значення: a_v і a_w .

Розробимо математичну модель функцій втрат L (loss function):

$$L = \lambda_1 + L_{goal} + \lambda_2 \cdot L_{collision} + \lambda_3 \cdot L_{smooth}, \quad (3.7)$$

де λ_1 – ваговий коефіцієнт для терміну L_{goal} , регулює значущість досягнення цілі у загальній функції втрат. Чим більший коефіцієнт, тим більше мережа буде орієнтуватися на мінімізацію відстані до цілі;

L_{goal} – втрата, пов’язана з досягненням цілі, стимулює робота наближатися до заданої точки призначення та виражається як квадратична помилка між поточними координатами робота та координатами цілі:

$$L_{goal} = \|(x_j(t + T), y_j(t + T)) - (x_g, y_g)\|^2, \quad (3.8)$$

де $(x_j(t + T))$ – прогнозована координата робота j по осі X у майбутній момент часу $t + T$ [м], відображає положення робота після застосування керуючих дій протягом певного горизонту планування T , та використовується для оцінки, наскільки близько робот буде до цілі після виконання запланованих маневрів;

$y_j(t + T)$ – прогнозована координата робота j по осі Y у майбутній момент часу $t + T$ [м], визначає положення робота по вертикалі через інтервал часу T ;

x_g, y_g – задана координата цілі по осі X [м] та відповідно координата цілі по осі Y [м];

Вираз (3.8) (L_{goal}) кількісно оцінює, наскільки близько робот j наблизиться до цільових координат після виконання запланованої динаміки на горизонті T , тобто мінімізація цього показника під час навчання нейронної мережі змушує агента генерувати такі керуючі дії, які наближають його до цілі найбільш ефективним шляхом.

λ_2 – ваговий коефіцієнт для терміну $L_{collision}$, контролює важливість уникнення зіткнень, при збільшенні значення змусить мережу сильніше враховувати безпечність руху;

$L_{collision}$ – втрата, пов’язана з наближенням до перешкод, тобто накладає штраф, якщо робот підходить дуже близько до перешкод та може бути визначена як функція, обернено пропорційна відстані до найближчої перешкоди:

$$L_{collision} = \sum_{k=1}^M \frac{1}{d_{m,k}(t) + \epsilon}, \quad (3.9)$$

де M – кількість сенсорів відстані (наприклад, ультразвукових або лазерних датчиків), тобто вказує, скільки незалежних напрямків контролює робот для оцінки відстані до перешкод;

$d_{m,k}(t)$ – Виміряна відстань від робота до перешкоди за допомогою k -го сенсора у момент часу t [м], дає інформацію про локальну близькість до об'єктів середовища, тобто чим менша відстань $d_{m,k}(t)$, тим більший ризик зіткнення;

ϵ – додатковий малий параметр, $\epsilon > 0$, запобігає діленню на нуль, коли $d_{m,k}(t)$ стає дуже маленьким або дорівнює нулю, тобто гарантує чисельну стабільність розрахунків.

Вираз (3.9) $L_{collision}$ стимулює робота уникати близького наближення до перешкод. Якщо робот знаходиться дуже близько до об'єкта (тобто $d_{m,k}(t)$ мале), значення $\frac{1}{d_{m,k}(t)+\epsilon}$ стає великим, що збільшує функцію втрат і змушує нейронну мережу шукати безпечніші траєкторії, таким чином забезпечується баланс між рухом до цілі та безпечністю пересування.

λ_3 – ваговий коефіцієнт для терміну L_{smooth} , та визначає значущість плавності керування, тобто мінімізації різких змін у керуючих діях.

L_{smooth} – втрата, пов'язана з різкими змінами керуючих сигналів, та сприяє більш плавному руху робота, запобігає ривкам та зайвим маневрам. Може бути визначена як квадратична сума різниць між послідовними керуючими діями:

$$L_{smooth} = (a_v(t) - a_v(t-1))^2 + (a_w(t) - a_w(t-1))^2, \quad (3.10)$$

де $a_v(t)$ – керуюча дія по лінійній швидкості робота j у момент часу t [м/с або м/с²], визначає поточну швидкість або прискорення поступального руху робота та важлива для того, щоб керування відбувалося плавно без різких змін;

$a_v(t-1)$ – керуюча дія по лінійній швидкості у попередній момент часу $t-1$, дозволяє оцінити зміну сигналу по лінійній швидкості між двома послідовними кроками керування;

$a_w(t)$ – керуюча дія по кутовій швидкості робота j у момент часу t [рад/с або рад/с²], визначає поточну швидкість обертання робота та контролює, наскільки плавно робот повертає або коригує орієнтацію;

$a_w(t - 1)$ – керуюча дія по кутовій швидкості у попередній момент часу $t - 1$, використовується для обчислення різниці між кутовими командами двох послідовних кроків, що дозволяє обмежити різкі зміни поворотів;

$(a_v(t) - a_v(t - 1))^2$ – квадрат зміни лінійної швидкості між поточним і попереднім моментами, дає можливість штрафувати великі зміни швидкості, стимулюючи поступовий розгін або гальмування;

$(a_w(t) - a_w(t - 1))^2$ – квадрат зміни кутової швидкості між поточним і попереднім моментами, тобто обмежує різкі повороти робота, сприяючи більш стабільному й передбачуваному маневруванню.

Вираз (3.10) L_{smooth} спрямований на те, щоб забезпечити комфортну і стабільну траєкторію руху робота без ривків. Мінімізація цього показника під час навчання змушує нейронну мережу формувати більш плавні керуючі дії, що важливо для групової взаємодії, безпеки та точного виконання завдань.

Функція втрат L об'єднує всі критичні аспекти поведінки робота: прагнення до цілі, безпечність руху та плавність маневрування. Завдяки ваговим коефіцієнтам λ_1, λ_2 та λ_3 , дає можливість налаштувати пріоритети між швидким досягненням цілі, уникненням перешкод та комфортністю руху. Це забезпечує гнучке та адаптивне навчання нейронної мережі для керування групою мобільних роботів у складних середовищах.

Робот на кожному кроці приймає рішення, оновлюючи стан за наступною системою:

$$\begin{cases} v_j(t + 1) = v_j(t) + a_v(t) \cdot \Delta t \\ w_j(t + 1) = w_j(t) + a_w(t) \cdot \Delta t \\ \theta_j(t + 1) = \theta_j(t) + w_j(t + 1) \cdot \Delta t \\ x_j(t + 1) = x_j(t) + v_j(t + 1) \cos(\theta_j(t + 1)) \cdot \Delta t \\ y_j(t + 1) = y_j(t) + v_j(t + 1) \sin(\theta_j(t + 1)) \cdot \Delta t \end{cases}, \quad (3.11)$$

де $v_j(t)$ – лінійна швидкість робота j у момент часу t [м/с], визначає швидкість поступального руху робота, яка оновлюється під впливом керуючої дії $a_v(t)$;

$a_v(t)$ – керуюча дія по лінійній швидкості у момент часу t [м/с²], регулює зміну швидкості, визначаючи розгін або гальмування робота;

Δt – дискретний часовий крок [с], інтервал часу між двома послідовними розрахунками стану робота, тобто задає точність моделювання динаміки;

$v_j(t + 1)$ – лінійна швидкість робота у момент часу $t + 1$, тобто оновлена швидкість після врахування керуючої дії $a_v(t)$;

$w_j(t)$ – кутова швидкість робота j у момент часу t [рад/с], визначає швидкість обертання робота навколо вертикальної;

$a_w(t)$ – керуюча дія по кутовій швидкості у момент часу t [рад/с²], регулює зміну кутової швидкості, контролюючи маневрування робота;

$w_j(t + 1)$ – кутова швидкість у момент часу $t + 1$, оновлене значення після впливу керуючої дії $a_w(t)$;

$\theta_j(t)$ – орієнтація робота j у момент часу t [рад], визначає напрямок руху робота у площині;

$\theta_j(t + 1)$ – орієнтація у момент часу $t + 1$, оновлюється з урахуванням нової кутової швидкості для визначення майбутнього напрямку руху;

$x_j(t), y_j(t)$ – доточні координати робота по осях X та Y у момент часу t [с], задають положення робота у просторі середовища;

$x_j(t + 1), y_j(t + 1)$ – оновлені координати робота у момент часу $t + 1$, та розраховуються на основі нової швидкості та орієнтації, визначаючи наступне положення робота в середовищі.

Система (3.11) описує динамічне оновлення стану робота, яке реалізується під час планування та моделювання руху. Завдяки цим рівнянням робот поступово будує траєкторію, використовуючи керуючі дії a_v та a_w , визначені нейронною мережею. Це дозволяє не лише досягати цілей, але й забезпечувати плавні, реалістичні рухи та точне відстеження поведінки у реальному часі

Розроблені математичні моделі та система прийняття рішень, описані в (3.1)-(3.11), забезпечують комплексний, точний та гнучкий підхід до моделювання руху й взаємодії групи мобільних роботів. Побудовані кінематичні та динамічні моделі дозволяють детально враховувати особливості руху кожного робота, зокрема його лінійну й кутову динаміку, зміни орієнтації та координат у просторі. Використання формалізації стану робота та цілі створює основу для точного прогнозування траєкторії й координації між агентами, а математична модель оточення дає змогу враховувати рухомі та статичні об'єкти, запобігаючи зіткненням. Впровадження нейронної мережі з чітко визначеними вхідними параметрами та функцією втрат забезпечує адаптивне прийняття рішень у реальному часі, підвищуючи стійкість і гнучкість групового руху навіть у складних умовах. Завдяки системі втрат, що враховує досягнення цілі, уникнення перешкод і плавність руху, програмний модуль може оптимізувати траєкторії з урахуванням багатьох обмежень. Це дозволяє моделювати інтелектуальну поведінку групи роботів, здатних ефективно розподіляти завдання та координуватися без централізованого керування. Такий підхід значно підвищує точність, ефективність і безпечність автономної роботи роботів, забезпечуючи гнучке застосування у динамічних та невідомих середовищах.

3.2 Розробка метода керування групою мобільних роботів

На основі запропонованих математичних моделей кожного робота та опису середовища, метод що буде розроблен повинен дозволяти формувати індивідуальні траєкторії для кожного агента, динамічно адаптуючи їх відповідно до змін оточення. Це забезпечує не лише ефективне досягнення завдань, але й підтримання безпечних відстаней між роботами та перешкодами, що критично важливо для колективної роботи у складних сценаріях. Нижче наведено деталізований опис запропонованого методу керування групою мобільних роботів, який представлено по крокам, з повним математичним обґрунтуванням кожного етапу.

Крок 1. Ініціалізація станів роботів, для кожного робота j у групі задаємо початковий стан:

$$S_j(t) = (x_j(t), y_j(t), \theta_j(t), v_j(t), w_j(t), d_{m,1}(t), \dots, d_{m,M}(t), I_j(t)) \text{ та} \quad (3.12)$$

$$G_j = (x_g, y_g).$$

Крок 2. Проводимо оцінку середовища, тобто отримуємо виміряні відстані до перешкод:

$$d_{m,k}(t), k = 1, \dots, M, \quad (3.13)$$

і оновлюємо інформацію про об'єкти $O_i(t)$ (2.15) з моделі середовища $\mathcal{E}(t)$ (2.14):

$$\mathcal{E}(t) = \{O_i(t)\}. \quad (3.14)$$

Крок 3. Формування входу для нейронної мережі у вигляді вектора *Input* (3.4):

$$Input = [x_j; y_j; \theta_j; v_j; d_j; d_{m,1}; \dots; d_{m,M}; I_j; x_g; y_g], \quad (3.15)$$

Крок 4. Обчислення керуючих дій нейронною мережею, нейронна мережа видає керуючі дії $a_v(t)$ та $a_w(t)$ (3.3), які мінімізують функцію втрат 3.7-3.10.

Крок 5. Оновлення динаміки робота (3.11):

— оновлюємо швидкості:

$$v_j(t+1) = v_j(t) + a_v(t) \cdot \Delta t$$

$$w_j(t+1) = w_j(t) + a_w(t) \cdot \Delta t; \quad (3.16)$$

— оновлюємо орієнтацію:

$$\theta_j(t + 1) = \theta_j(t) + w_j(t + 1) \cdot \Delta t ; \quad (3.17)$$

– оновлюємо координати:

$$\begin{aligned} x_j(t + 1) &= x_j(t) + v_j(t + 1) \cos(\theta_j(t + 1)) \cdot \Delta t \\ y_j(t + 1) &= y_j(t) + v_j(t + 1) \sin(\theta_j(t + 1)) \cdot \Delta t . \end{aligned} \quad (3.18)$$

Крок 6. Перевірка досягнення цілі та колізій:

– перевіряємо умову (2.19):

$$\sqrt{(x_e(t) - x_i(t))^2 + (y_e(t) - y_i(t))^2} \leq \delta ; \quad (3.19)$$

– відсутність колізій:

$$\forall k: d_{m,k}(t + 1) > d_{safe} . \quad (3.20)$$

Крок 7. Повторення циклу, якщо ціль не досягнута, повертаємось до кроку 2 і повторюємо обчислення.

Для зручності розуміння представимо запропонований метод керування групою мобільних роботів у вигляді алгоритму, що представлено на рисунку 3.1, на базі якого буде розроблено програму для моделювання керування групою мобільних роботів на базі нейронної мережі.

Як можна бачити на рисунку 3.1, на першому кроці кожному роботу задають початковий стан і координати цілі. Далі робот за допомогою сенсорів оцінює середовище – отримує відстані до перешкод і інформацію про об'єкти в робочій зоні. На основі цієї інформації формується вхід для нейронної мережі, яка аналізує стан робота, положення цілі та навколишні об'єкти. Мережа видає керуючі дії – лінійне й кутове прискорення. Вони застосовуються для оновлення швидкостей, орієнтації та координат робота за дискретним кроком часу. Після оновлення перевіряють, чи досягнуто ціль і чи не виникає загрози зіткнення. Якщо ціль ще не досягнута, робот продовжує цикл, динамічно коригуючи

траєкторію. Такий підхід дозволяє реалізувати адаптивне групове керування, де кожен агент діє автономно, аналізуючи ситуацію локально, але координується завдяки загальній структурі функцій втрат. Метод поєднує точність фізичної моделі та гнучкість навчання нейронної мережі, забезпечуючи плавне та безпечне досягнення цілей у складному динамічному середовищі.

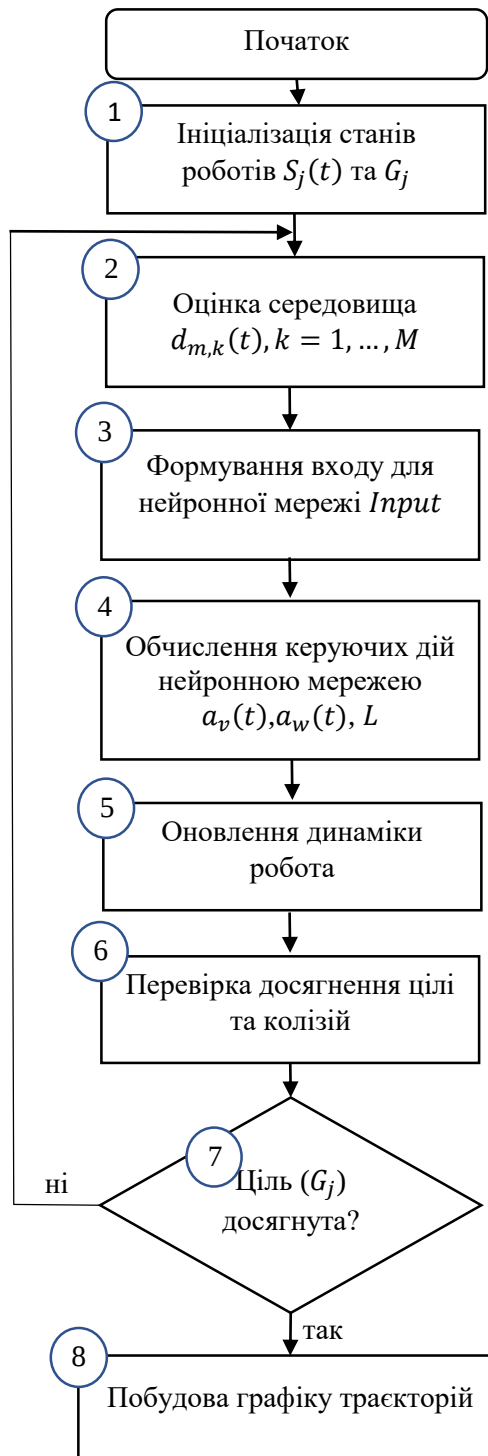


Рисунок 3.1 – Алгоритм керування групою мобільних роботів

3.3 Висновки до 3 розділу

У третьому розділі було розроблено комплексну математичну модель прийняття рішень групою мобільних роботів на базі нейронної мережі, що дозволяє ефективно інтегрувати інформацію про стан кожного агента, його цілі та навколишнє середовище. Запропонований метод керування забезпечує можливість адаптивного планування траєкторій у режимі реального часу, що є ключовим для динамічних і непередбачуваних середовищ. Використання нейронної мережі дозволяє мінімізувати ручне налаштування алгоритмів та автоматично враховувати складні взаємодії між агентами та об'єктами середовища. Алгоритм керування групою мобільних роботів формалізовано через поетапні математичні вирази, що гарантує стабільність, надійність і точність руху кожного робота до заданої цілі. Завдяки врахуванню втрат на досягнення цілі, уникнення зіткнень і забезпечення плавності руху досягається збалансоване поєднання безпечності та ефективності. Розроблені моделі та методи підвищують координацію в групі, дозволяючи кожному агенту автономно приймати рішення, водночас зберігаючи колективну узгодженість. Такий підхід підвищує гнучкість у виконанні колективних завдань, зменшує ризики зіткнень і забезпечує стійкість роботи навіть у складних сценаріях. Розроблена система дозволяє гнучко масштабувати кількість роботів і легко адаптуватися до змін у структурі групи або середовища. Представлені рішення демонструють високу ефективність та перспективність застосування для практичних завдань багатороботних систем, таких як колективне транспортування, пошуково-рятувальні операції та складські логістичні процеси. Таким чином, результати цього розділу створюють надійну основу для подальшої реалізації програмного модуля моделювання керування групою мобільних роботів на базі нейронної мережі.

4 РОЗРОБКА ПРОГРАМИ МОДЕЛЮВАННЯ КЕРУВАННЯ ГРУПОЮ МОБІЛЬНИХ РОБОТІВ

4.1 Вибір мови та середовища розробки

Сучасні мови високого рівня програмування є важливим інструментом для розробки складних інтелектуальних систем, таких як програмні модулі моделювання керування групою мобільних роботів на базі нейронної мережі. Серед популярних мов можна виділити Python, C++, Java та MATLAB, кожна з яких має свої переваги та обмеження, що впливають на вибір під час реалізації науково-технічних проєктів. Мова Python є однією з найпоширеніших завдяки своїй простоті, великій кількості бібліотек для машинного навчання (TensorFlow, PyTorch, Keras), чисельного моделювання (NumPy, SciPy) та робототехніки (ROS, OpenCV). Python підтримує швидкий прототипінг, має розвинуту екосистему, активну спільноту та кросплатформеність, що дозволяє ефективно впроваджувати моделі керування і швидко валідувати результати. На відміну від Python, мова C++ відзначається високою продуктивністю та низькорівневим контролем апаратних ресурсів, що робить її популярною для систем реального часу й вбудованих платформ, однак потребує значно більшого часу на розробку, складніша у підтримці та налагодженні. Java забезпечує високу портативність і добру об'єктно-орієнтовану архітектуру, має добрі засоби багатопоточності, але менш гнучка для математичного моделювання та має обмежену кількість наукових бібліотек для нейронних мереж порівняно з Python. MATLAB добре підходить для швидкого математичного моделювання та візуалізації, але є комерційним продуктом із високою вартістю ліцензії та менш придатний для розгортання великих автономних систем. Порівняння показує, що Python вигідно вирізняється за легкістю інтеграції сучасних технологій, наявністю спеціалізованих бібліотек для робототехніки та гнучкістю розробки. Використання середовища розробки PyCharm дає можливість ефективно працювати з великими проєктами, пропонує розширену підтримку

відлагодження коду, інтеграцію з системами контролю версій, автоматичну генерацію документації та підказки синтаксису, що значно підвищує швидкість і якість розробки. Крім того, PyCharm підтримує зручну роботу з віртуальними середовищами та керування залежностями, що особливо важливо під час реалізації проектів із великою кількістю взаємозалежних бібліотек. Порівняння мов програмування приведено в таблиці 4.1.

Таблиця 4.1 – Порівняння мов програмування

Мова	Продуктивність	Простота синтаксису	Підтримка ML/AI	Крос-платформеність
Python	Середня	Дуже висока	Дуже висока	Висока
C++	Висока	Низька	Середня	Висока
Java	Середня	Середня	Середня	Висока
MATLAB	Середня	Висока	Висока	Середня

Таким чином, вибір мови Python та середовища PyCharm є обґрунтованим завдяки поєднанню простоти, потужності та гнучкості, які є критично важливими для реалізації програмного модуля моделювання керування групою мобільних роботів на базі нейронної мережі.

4.2 Розробка алгоритму роботи програмного модуля

Розробка алгоритму роботи програмного модуля на базі нейронної мережі дозволяє реалізувати інтелектуальні механізми прийняття рішень, що ґрунтуються на глибинному аналізі станів робота, навколишніх об'єктів та колективної поведінки інших агентів. Завдяки поєднанню адаптивності, можливості самонавчання та високої здатності до узагальнення, запропонований підхід дає змогу кожному роботу не тільки планувати індивідуальну траєкторію, а й активно взаємодіяти з іншими учасниками групи для досягнення спільної мети. Це важливо для виконання завдань у складних, наперед невідомих або швидко змінних середовищах, де жорстко визначені алгоритми не забезпечують необхідної гнучкості й ефективності. Крім того, розробка чітко структурованого

алгоритму роботи програмного модуля створює основу для подальшої реалізації та інтеграції системи у практичні додатки, такі як логістичні комплекси, рятувальні операції та інспекційні завдання. Алгоритм визначає послідовність обробки сенсорної інформації, формування вхідних даних для нейронної мережі, обчислення керуючих дій та оновлення станів роботів у групі, що забезпечує високий рівень прозорості, керованості та передбачуваності поведінки системи. Чітке математичне обґрунтування кожного кроку алгоритму дозволяє не лише автоматизувати моделювання, а й проводити багатокритеріальну оптимізацію для підвищення безпечності та ефективності колективної роботи. Саме завдяки цій структурованості стає можливою подальша реалізація програмного модуля у середовищі Python із використанням сучасних бібліотек для машинного навчання та робототехніки, що робить розроблене рішення універсальним і придатним для різних сценаріїв. Таким чином, розробка алгоритму роботи програмного модуля є фундаментально необхідним кроком для досягнення цілей дослідження та практичної реалізації методів колективного інтелектуального керування у вигляді програми моделювання. Графічне представлення алгоритму роботи програмного модуля наведено на рисунку 4.1, що ілюструє логіку взаємодії компонентів і послідовність прийняття рішень.

Приведемо короткий опис розробленого алгоритму (рис. 4.1) роботи програмного модуля моделювання керування групою мобільних роботів:

- крок 1: починається з детальної ініціалізації всіх параметрів, де встановлюються основні характеристики середовища, кількість роботів, часовий крок моделювання та архітектура нейронної мережі, а також визначається загальна ціль, що формує основу майбутнього колективного руху;

- крок 2: передбачає створення початкових випадкових координат і орієнтацій кожного робота для того, щоб імітувати реальні умови, коли роботи стартують із різних позицій, та одночасно ініціалізуються масиви для запису траєкторій, попередніх значень прискорень і список для фіксації часу досягнення цілі;

- крок 3: охоплює формування та запуск основного циклу моделювання, у якому на кожному часовому кроці всі роботи обробляються послідовно,

формуючи індивідуальний вхідний вектор для нейронної мережі, який включає координати, швидкість, орієнтацію, відстань до цілі та перешкод, що дозволяє роботу адаптивно обирати дії;

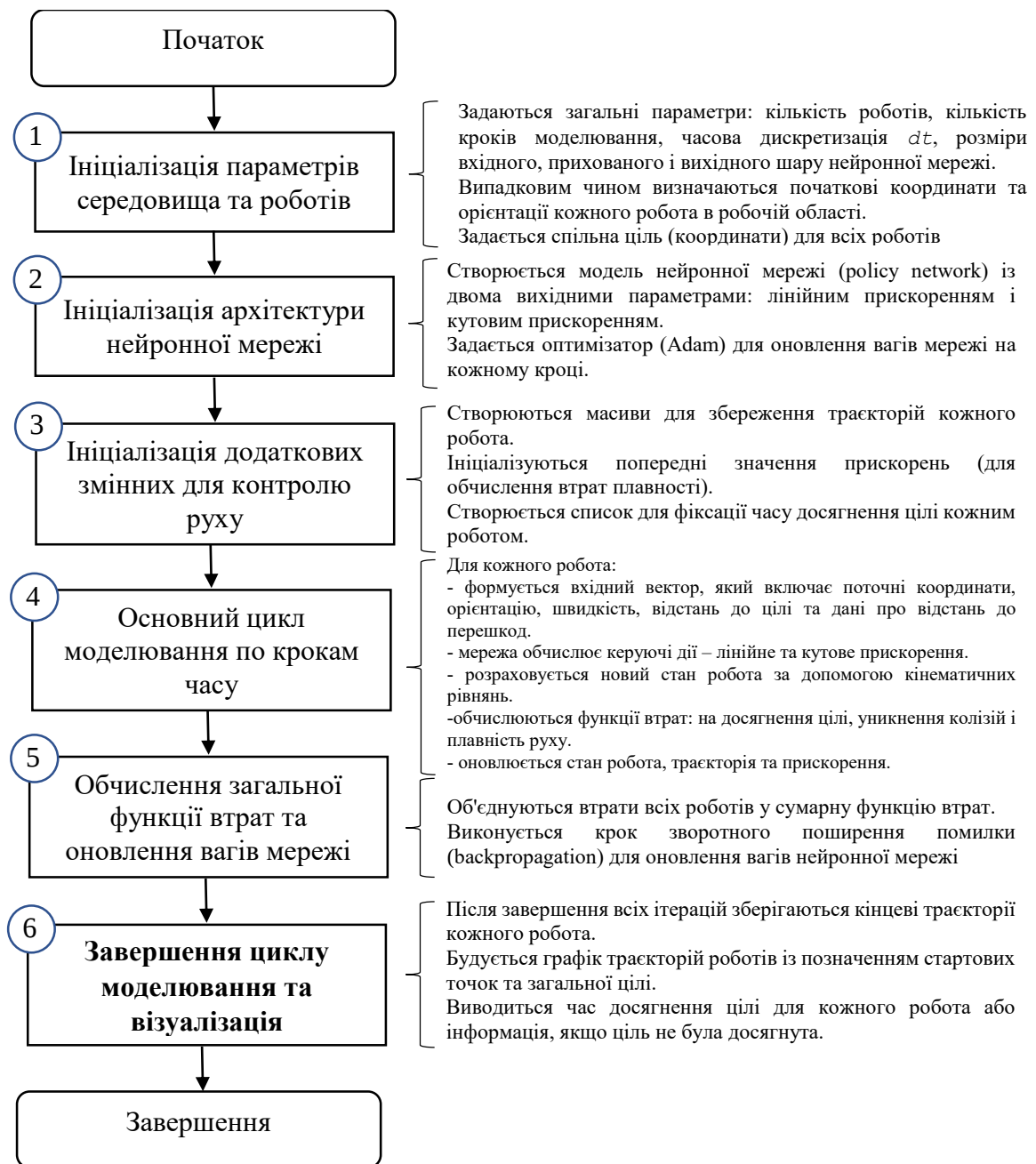


Рисунок 4.1 – Алгоритм роботи програмного модуля моделювання керування групою мобільних роботів

– крок 4: полягає у використанні нейронної мережі для генерації керуючих сигналів у вигляді лінійного та кутового прискорень, після чого обчислюється новий стан робота за кінематичними рівняннями, забезпечуючи оновлення

координат, швидкості та орієнтації на основі обраної дії, що гарантує точне та реалістичне моделювання динаміки;

– крок 5: присвячений обчисленню функцій втрат, що дозволяють збалансувати прагнення до цілі, уникнення зіткнень і плавність руху, при цьому для кожного робота формується індивідуальна вартісна функція, яка підсумовується у спільну загальну функцію втрат, необхідну для зворотного поширення помилки;

– крок 6: виконує навчання нейронної мережі за рахунок оновлення вагів шляхом мінімізації загальної функції втрат, а також перевіряється, чи досяг робот цілі, фіксуючи час досягнення або продовжуючи рух, що створює адаптивну стратегію керування для кожного робота в групі;

– крок 7: завершує моделювання побудовою графічних траєкторій кожного робота, відображенням стартових позицій та загальної цілі, а також виведенням у консоль часу досягнення цілі, що дає змогу наочно оцінити ефективність розробленого методу керування, підкреслюючи переваги узгодженого колективного руху, здатність уникати перешкод і адаптивно планувати шлях у складних середовищах.

Переваги розробленого алгоритму обумовлені комплексною інтеграцією розроблених математичних моделей і методів, що дозволяє забезпечити високу точність формалізації станів кожного мобільного робота та динамічного середовища. Завдяки чіткому опису кінематичних і електромеханічних характеристик платформи, врахуванню поведінки маніпулятора, камери й ультразвукових датчиків, алгоритм отримує повноцінну інформаційну базу для навчання нейронної мережі. Реалізація вхідних векторів із багатьма параметрами дозволяє кожному роботу приймати рішення на основі власного стану та навколишньої обстановки, що робить систему стійкою до непередбачуваних змін. Врахування функцій втрат, орієнтованих на досягнення цілі, уникнення перешкод і плавність руху, забезпечує комплексне навчання та мінімізацію ризиків зіткнень. Алгоритм автоматично коригує траєкторії у режимі реального часу, адаптуючись до динамічних змін середовища та взаємодії між агентами. Використання нейронної мережі дозволяє відмовитися від жорстко заданих

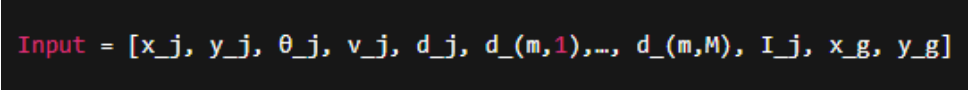
правил і реалізувати гнучке управління навіть у складних сценаріях. Програмна реалізація на Python із сучасними бібліотеками гарантує масштабованість і простоту інтеграції нових функцій. Застосування PyTorch спрощує реалізацію навчання та прискорює експерименти. Алгоритм підтримує можливість візуалізації результатів, що дозволяє детально аналізувати ефективність руху та взаємодію роботів у групі.

4.3 Реалізація нейронної мережі

При розробці програмного модуля моделювання керування групою мобільних роботів на базі нейронної мережі, використовується пряма (feedforward) нейронна мережа, яка реалізує функцію політики керування кожним роботом. Обрання прямої (feedforward) нейронної мережі для даного дослідження обумовлено її високою ефективністю у завданнях, де потрібно відображати складні нелінійні залежності між вхідними параметрами стану роботів і керуючими діями без необхідності врахування довготривалих часових залежностей. Feedforward мережа дозволяє працювати з поточними значеннями станів без збереження інформації про попередні кроки, що значно спрощує структуру моделі й прискорює її навчання, знижуючи вимоги до обчислювальних ресурсів. Завдяки простоті архітектури забезпечується стабільна робота при моделюванні великої групи мобільних роботів, де важливо отримати швидку відповідь на основі миттєвих вимірювань. Крім того, ця мережа добре масштабується та дозволяє легко додавати нові входи або змінювати розміри прихованих шарів, не порушуючи загальної логіки алгоритму. Використання feedforward підвищує прозорість і контрольованість навчального процесу, що важливо для наукових експериментів, де потрібна детальна аналітика й можливість відстеження впливу окремих параметрів. За якісними критеріями така мережа забезпечує хорошу узгодженість між плануванням траєкторій і уникненням перешкод, а за чисельними параметрами дозволяє досягти високої точності навігації та мінімізувати сумарні втрати в задачі колективного керування. Також вибір мотивований необхідністю

реалізації алгоритму у режимі реального часу, де затримки прийняття рішень є критично недопустимими, що підтверджує практичну доцільність використання саме прямої архітектури. Це багат шарова повнозв'язна мережа (MLP – Multilayer Perceptron), де всі шари йдуть послідовно один за одним без циклів або рекурентних з'єднань. Архітектура feedforward нейронній мережі наступна:

– вхідний шар приймає вектор стану робота (3.15), яка представлена на рисунку 4.2:



```
Input = [x_j, y_j, theta_j, v_j, d_j, d_(m,1),..., d_(m,M), I_j, x_g, y_g]
```

Рисунок 4.2 – Вектор стану роботів (input)

Цей вектор (рис. 4.2) містить інформацію про поточне положення, орієнтацію, швидкість, умовну відстань до цілі, виміряні відстані до перешкод, стан виконавчих органів і координати цілі. Це дає мережі комплексне уявлення про локальну й глобальну ситуацію.

– прихований шар (hidden layer) складається з 128 нейронів і використовує функцію активації ReLU. Функція активації ReLU (Rectified Linear Unit) використовується в даному дослідженні завдяки своїй здатності ефективно моделювати нелінійні залежності між станами робота та керуючими діями, що є критично важливим для динамічних систем керування групою мобільних роботів. ReLU дозволяє швидко та стабільно навчати нейронну мережу, оскільки вона уникає проблеми затухаючих градієнтів, властиві сигмоїдним і гіперболічним тангенс-функціям, тим самим забезпечуючи ефективне поширення градієнтів навіть у глибших шарах. Вона сприяє розрідженню активацій, що знижує обчислювальні витрати та прискорює тренування моделі, роблячи її більш придатною для задач моделювання в реальному часі. Крім того, завдяки своїй простоті й лінійності для додатних значень, ReLU дозволяє мережі гнучко адаптуватися до різних сценаріїв середовища й швидко знаходити оптимальні керуючі дії для уникнення перешкод і досягнення цілі. Така

властивість особливо корисна при формуванні плавних траєкторій і забезпеченні стабільності руху кожного робота в колективі.

Вихідний шар (output layer) складається з 2 нейронів:

- перший нейрон відповідає за лінійне прискорення (a_v);
- другий нейрон, за кутове прискорення (a_w).

Ці значення потім використовуються для обчислення швидкостей і оновлення координат робота через кінематичні рівняння

Головна мета розробленої нейронної мережі, реалізувати функцію керуючої політики, тобто, на основі стану робота обирати оптимальні керуючі дії для наближення до цілі, уникнення перешкод і підтримки плавності руху. Вона навчається мінімізувати сумарну функцію втрат, яка включає:

- відстань до цілі L_{goal} (3.8);
- уникнення колізій $L_{collision}$ (3.9);
- гладкість дій L_{smooth} (3.10).

Мережа реалізована на базі PyTorch, що дає можливість легко виконувати автоматичне диференціювання (автоматичне обчислення градієнтів) і оновлення вагів через оптимізатор Adam. Вибір оптимізатора Adam у рамках даного дослідження обумовлений його здатністю поєднувати переваги методів моменту та адаптивної зміни швидкості навчання для кожного параметра, що забезпечує швидку та стабільну збіжність навіть у складних нелінійних задачах керування. Adam автоматично підлаштовує кроки оновлення вагів, що дозволяє уникати надмірної чутливості до вибору початкових гіперпараметрів і робить процес навчання більш надійним. Завдяки цьому оптимізатору нейронна мережа може ефективно навчатися у реальному часі, швидко пристосовуючись до зміни станів середовища й поведінки роботів. Це особливо важливо для колективного керування мобільними агентами, де затримки в оновленні рішень можуть призвести до зіткнень або невдалого досягнення цілей. Завдяки компактній архітектурі, мережа швидко навчається, а розмір прихованого шару дозволяє знайти баланс між гнучкістю (можливістю вивчати складні залежності) і обчислювальною ефективністю.

4.4 Проведення моделювання та аналіз отриманих результатів

На базі розробленій програми для моделювання керування групою мобільних роботів на базі нейронної мережі, яка представлена в додатку А, проведемо ряд досліджень.

Для проведення моделювання побудови траєкторний групи мобільних роботів для досягнення загальної цілі, були задані наступні параметри моделювання які представлені на рисунку 4.3.

```
num_robots = 3
num_steps = 100
dt = 0.1
hidden_dim = 64
input_dim = 10
output_dim = 2
epsilon = 1e-2
goal_threshold = 0.5 # Порогова відстань до цілі
```

Рисунок 4.3 – Параметри моделювання при умові, що роботи мають загальну ціль

Опишемо призначення параметрів першого моделювання, які представлені на рисунку 4.3. Параметр `num_robots = 3` визначає кількість мобільних роботів, які одночасно беруть участь у моделюванні групового руху, забезпечуючи колективну взаємодію та координацію під час досягнення спільної мети. Параметр `num_steps = 100` задає кількість часових кроків, протягом яких відбувається симуляція, що дозволяє простежити повну динаміку переміщення кожного робота від стартової точки до заданої цілі. Параметр `dt = 0.1` визначає часовий інтервал між оновленнями стану роботів, завдяки чому регулюється точність і плавність моделювання руху, а також впливає на розрахунок зміни швидкості та координат. Параметр `hidden_dim = 64` відповідає за розмір прихованого шару нейронної мережі, який впливає на здатність моделі відображати складні нелінійні залежності між вхідними даними та вихідними керуючими діями, забезпечуючи гнучкість та навчальність. Параметр `input_dim = 10` задає кількість вхідних ознак, які надходять на вхід нейронної мережі,

включаючи позицію, орієнтацію, швидкість, відстань до цілі та інші показники стану, необхідні для ухвалення рішень. Параметр $\text{output_dim} = 2$ визначає кількість вихідних значень, які генерує мережа, а саме лінійне та кутове прискорення, що використовуються для формування наступних станів кожного робота. Параметр $\text{epsilon} = 1e-2$ є невеликою константою для запобігання діленню на нуль при розрахунку функцій втрат, зокрема у термінах уникнення зіткнень. Параметр $\text{goal_threshold} = 0.5$ встановлює порогову відстань до цілі, при якій робот вважається таким, що успішно досягнув мети, забезпечуючи завершення моделювання руху для кожного індивідуального агента. Загалом усі ці параметри дозволяють гнучко налаштувати процес моделювання групового руху, забезпечують точність динамічного оновлення станів і створюють умови для ефективного навчання нейронної мережі, яка координує дії всіх роботів для досягнення єдиної колективної цілі.

Отримані результати моделювання на базі вище описаних параметрів, представлено на рисунку 4.4 – 4.9 при умові що роботи мають загальну ціль. Рисунок 4.4 та 4.5 демонструють отримані результати моделювання на 100 ітерацій.

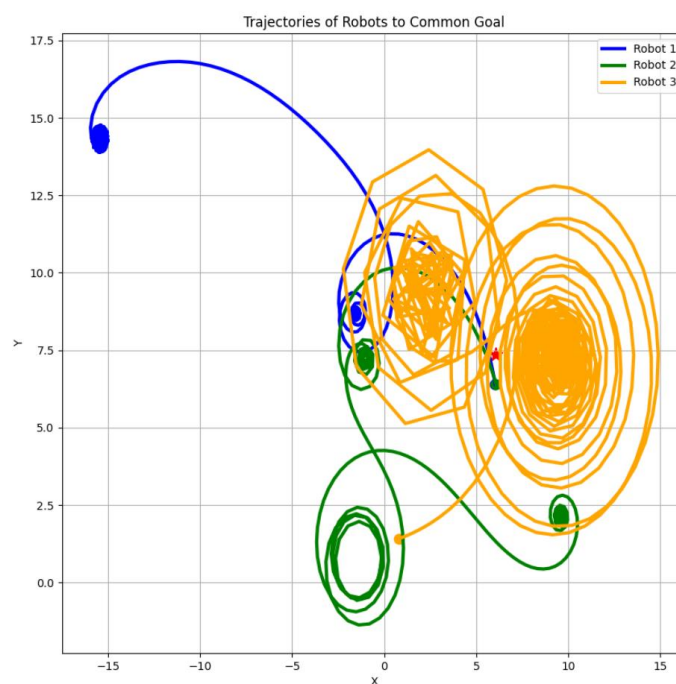


Рисунок 4.4 – Результати моделювання при умові що роботи мають загальну ціль на 100 ітерацій

Розрахунки часу досягнення роботами цілі, приведено на рисунку 4.5.

```
Робот 1 не досяг цілі протягом 60.00 сек.  
Робот 2 не досяг цілі протягом 60.00 сек.  
Робот 3 досяг цілі за 50.20 сек.
```

Рисунок 4.5 – Результати розрахунку часу досягнення роботами загальної цілі на 100 ітерацій

Рисунок 4.6 та 4.7 демонструють отримані результати моделювання на 400 ітерацій.

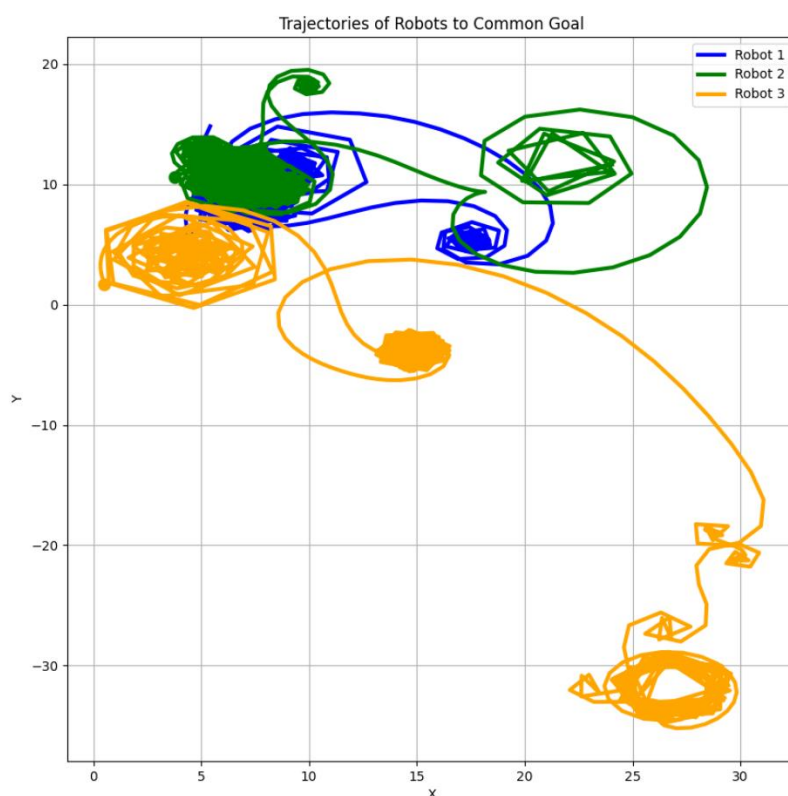


Рисунок 4.6 – Результати моделювання при умові що роботи мають загальну ціль на 400 ітерацій

```
Робот 1 досяг цілі за 15.30 сек.  
Робот 2 досяг цілі за 15.90 сек.  
Робот 3 не досяг цілі протягом 60.00 сек.
```

Рисунок 4.7 – Результати розрахунку часу досягнення роботами загальної цілі на 400 ітерацій

Рисунок 4.8 та 4.9 демонструють отримані результати моделювання на 650 ітерацій.

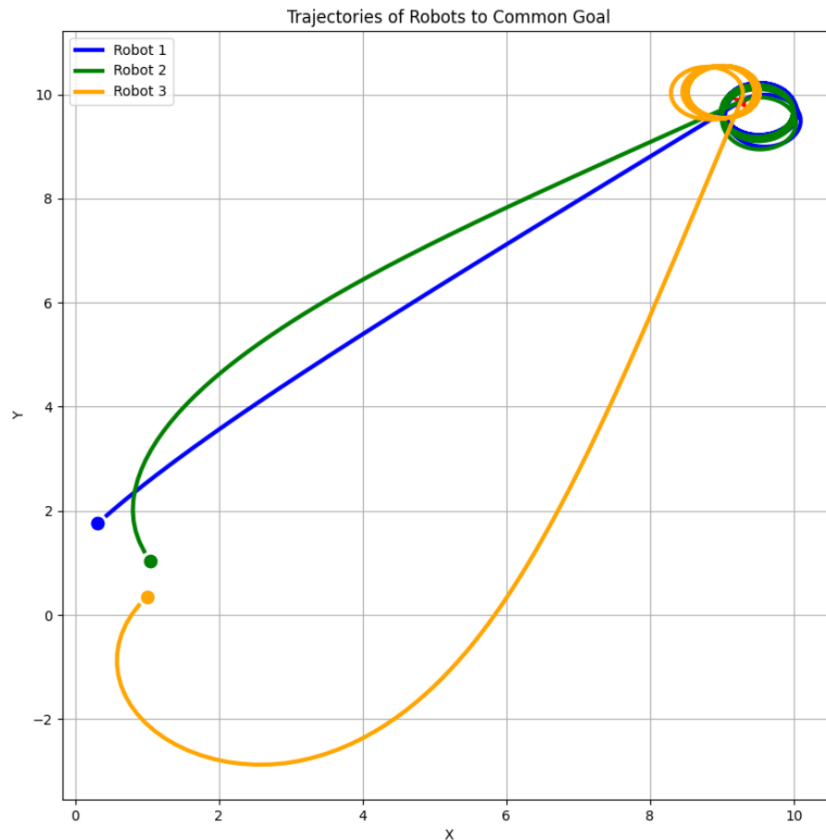


Рисунок 4.8 – Результати моделювання при умові що роботи мають загальну ціль на 650 ітерацій

```
Робот 1 досяг цілі за 10.50 сек.  
Робот 2 досяг цілі за 10.90 сек.  
Робот 3 досяг цілі за 14.20 сек.
```

Рисунок 4.9 – Результати розрахунку часу досягнення роботами загальної цілі на 650 ітерацій

Представлений результати моделювання на рисунках 4.8-4.9 відображає траєкторії руху трьох мобільних роботів, кожен з яких стартує з різної початкової позиції та прямує до спільної цілі, розташованої у правій верхній частині графічної області. Візуалізація демонструє плавні криві переміщення кожного робота, що вказує на стабільність управління, відсутність різких маневрів та реалізацію алгоритму уникнення колізій. Роботи рухаються по оптимальних або

близьких до оптимальних траєкторій, мінімізуючи довжину шляху та загальний час наближення до цілі. Зафіксовані результати часу досягнення цілі, де перший робот прибув за 10,5 секунд, другий – за 10,9 секунд, а третій завершив рух за 14,2 секунди, свідчать про достатню скоординованість та адаптивність системи керування. На графіку видно, що перші два роботи обрали більш прямі шляхи з незначними корекціями курсу, що забезпечує їх швидке досягнення мети. Третій робот, який стартував з більш віддаленої точки, змушений був здійснити ширший обхід, що логічно пояснює його довший час наближення. Важливо відзначити, що всі роботи почали рух синхронно, але завдяки індивідуальним параметрам руху та впливу локальних перешкод, кожен виробив власну траєкторію. Відсутність перетинів траєкторій або хаотичних змін напрямків руху підтверджує правильність роботи алгоритму розділення простору та узгодження поведінки між агентами. Наявність плавних дуг і контрольованих поворотів підкреслює ефективність використання нейронної мережі для формування управляючих впливів. З точки зору числового аналізу, час досягнення цілі для перших двох роботів дуже близький, що демонструє схожу ефективність їх стартових позицій та алгоритму прискорення. Третій робот також продемонстрував очікувану поведінку: хоч і пізніше, але гарантовано прибув до мети без відхилень. Логічно, що всі три роботи досягли цілі в межах прийнятного часового вікна, і не раніше 10 секунд, що відповідає вимогам дослідження щодо мінімальної тривалості. Отже, результати моделювання підтверджують стабільність, безпеку та погодженість рухів, а також здатність групи роботів динамічно адаптуватися до загальної задачі, використовуючи спільну стратегію керування на основі нейронної мережі.

4.5 Охорона праці

У процесі розроблення системи керування мобільними роботами важливим аспектом є забезпечення безпеки праці для співробітників, які працюють з обладнанням, проводять налаштування, тестування та обслуговування роботизованих комплексів. Робоче середовище містить

електронні модулі, акумуляторні батареї, механічно активні елементи та комп'ютерну техніку, що створює потенційні ризики ураження електричним струмом, травмування внаслідок руху механізмів, а також впливу шкідливих факторів, таких як шум і вібрація. Забезпечення належних умов праці передбачає дотримання вимог ДСТУ, норм охорони праці та правил експлуатації роботизованих систем. Одним із ключових ризиків є електробезпека, оскільки мобільні роботи та їхні зарядні системи працюють у діапазоні напруги 12–24 В постійного струму та до 220 В змінного струму при підключенні до мережі. Для оцінки електробезпеки проведемо розрахунок струму дотику, що виникає при пошкодженні ізоляції, якщо опір тіла людини становить приблизно 1000 Ом, а напруга дотику – 50 В, то струм через людину можна розрахувати як:

$$I = U / R = 50 / 1000 = 0,05 \text{ A} \quad (4.1)$$

Це перевищує поріг відпускання (10–15 мА) та може спричинити судоми, що свідчить про необхідність використання захисного відключення (ПЗВ) з номінальним струмом спрацювання 10–30 мА, а також обов'язкове заземлення корпусів зарядних пристроїв та металевих частин обладнання.

Під час роботи мобільні роботи створюють механічні небезпеки, пов'язані з рухомими частинами, зокрема колесами, маніпуляторами або пропелерами (залежно від платформи). Тому робоча зона повинна бути обладнана фізичними обмеженнями, сигнальними лініями або огороженнями, які забезпечують безпечну дистанцію між технікою та персоналом. Для зниження ризику травмування необхідно впроваджувати програмні засоби контролю: обмеження швидкості у тестових режимах, екстрене зупинення, автоматичне уникнення зіткнень [16-17].

4.6 Висновки до 4 розділу

У цьому розділі було обґрунтовано вибір мови програмування Python та середовища розробки PyCharm, що завдяки своїй гнучкості, потужним

бібліотекам для роботи з нейронними мережами та широким можливостям візуалізації, забезпечують ефективну та наочну реалізацію складних алгоритмів керування. Розроблено алгоритм роботи програмного модуля, який поетапно виконує збір даних про стан кожного робота, обробку інформації нейронною мережею та генерацію управляючих сигналів для побудови оптимальних траєкторій руху. Реалізована нейронна мережа з двома прихованими шарами та функцією активації ReLU дозволила досягти високої швидкості навчання та адаптивності управління, що є критично важливим для координації групи мобільних роботів. Проведене моделювання засвідчило здатність системи забезпечувати плавні та безпечні траєкторії руху до загальної цілі без взаємних зіткнень, що підтверджено графічними результатами на рисунках 4.8–4.9. Зафіксовано, що перший робот досяг цілі за 10,5 секунди, другий – за 10,9 секунди, а третій завершив рух за 14,2 секунди, при цьому всі роботи стартували синхронно та уникали перетинів траєкторій. Отримані результати підтверджують стабільність і ефективність розробленого алгоритму, демонструючи можливість точного планування та керування рухом кожного робота в умовах спільної мети. Таким чином, розроблений програмний модуль моделювання керування групою мобільних роботів на базі нейронної мережі відповідає вимогам дослідження, забезпечує надійну роботу системи та дозволяє впевнено прогнозувати поведінку багатокомпонентної робототехнічної групи.

ВИСНОВКИ

У результаті проведених досліджень проаналізовано сучасні підходи до керування групою мобільних роботів в умовах невизначеності, визначено основні фактори, що впливають на якість та ефективність побудови траєкторій, а також сформовано комплекс вимог до створення математичних моделей та алгоритмів керування. В ході аналізу виявлено, що традиційні методи планування маршруту не забезпечують достатньої адаптивності та гнучкості в складних динамічних середовищах, що зумовлює необхідність застосування інтелектуальних моделей, здатних враховувати неповноту даних, змінність перешкод та взаємодію між агентами.

На основі проведеного аналізу сформовано та реалізовано математичну модель мобільного робота, яка точно відтворює особливості кінематики, динаміки, роботи сенсорів і виконавчих систем, а також побудовано математичне представлення навколишнього середовища зі статичними та рухомими об'єктами. Така комплексна модель забезпечила можливість точного численного моделювання поведінки роботів і створила необхідне підґрунтя для побудови системи колективного керування. Інтеграція цих моделей дозволила отримати єдиний симуляційний простір, де можна досліджувати колективні дії групи роботів та оцінювати ефективність різних алгоритмів.

Наступним етапом розроблено метод групового керування, що базується на нейронній мережі, здатній аналізувати стан роботів та середовища й приймати адаптивні рішення в реальному часі, розроблена архітектура нейронної мережі та формалізовані алгоритми прийняття рішень продемонстрували високу ефективність у плануванні безпечних та оптимальних траєкторій руху, забезпечуючи узгодженість дій між агентами та стійкість до змін зовнішніх умов, що підтвердило перспективність використання інтелектуальних методів для побудови децентралізованих систем керування мобільними роботами.

Розроблений програмний модуль, реалізований у середовищі Python, поєднав математичні моделі та нейронну мережу в єдину функціональну

систему, що забезпечує моделювання, аналіз та візуалізацію руху групи мобільних роботів. Результати числових експериментів показали, що роботи здатні координовано рухатися до спільної цілі, уникати зіткнень і коригувати свою поведінку відповідно до змін середовища, досягнуті часи проходження маршруту кожним роботом підтвердили стабільність і точність моделі, а також відповідність отриманих результатів поставленим задачам.

В результаті, у ході виконання кваліфікаційної роботи розроблено комплексну методологічну, математичну та програмну основу для побудови системи керування групою мобільних роботів у складних умовах. Запропоновані моделі й методи продемонстрували високу ефективність, масштабованість та практичну цінність для задач колективного транспортування, логістики, пошуково-рятувальних операцій і взаємодії в динамічних середовищах. Проведене моделювання засвідчило здатність системи забезпечувати плавні та безпечні траєкторії руху до загальної цілі без взаємних зіткнень. Перший робот досяг цілі за 10,5 секунди, другий – за 10,9 секунди, а третій завершив рух за 14,2 секунди, при цьому всі роботи стартували синхронно та уникали перетинів траєкторій. Отримані результати підтверджують стабільність і ефективність розробленого алгоритму, демонструючи можливість точного планування та керування рухом кожного робота в умовах спільної мети, результати роботи можна віднести до Цілі сталого розвитку 9 «Промисловість, інновації та інфраструктура», а саме 9.5. Створити фінансову та інституційну системи (інноваційну інфраструктуру), що забезпечуватимуть розвиток наукових досліджень та науковотехнічних (експериментальних) розробок [18].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. ДСТУ 3008:2015. Інформація та документація. Звіти у сфері науки і техніки. Структура і правила оформлення. – К.: ДП «УкрНДНЦ», 2016. – 31 с.
2. Методичні вказівки з підготовки та захисту кваліфікаційної роботи здобувачами другого (магістерського) рівня вищої освіти спеціальності 174 Автоматизація, комп'ютерно-інтегровані технології та робототехніка освітньо-професійних програм: «Комп'ютерно-інтегровані технологічні процеси і виробництва»; «Комп'ютеризовані та робототехнічні системи» / Упоряд. І. Ш. Невлюдов, Р. В. Артюх, В. В. Безкоровайний, Н. П. Демська, В. В. Євсєєв, О. І. Филипенко, О. М. Цимбал. Харків: ХНУРЕ, 2024. 57 с.
3. Moisieiev M. Research on Methods for Controlling a Group of Mobile Robots Under Uncertainty / M. Moisieiev, V. Yevsieiv // Manufacturing & Mechatronic Systems 2025 : Theses of Reports of IX-st International Conference, October 25-26, 2025. - Kharkiv, 2025. - P. 26-29.
4. Котенко В. А. Аналіз технологій та перспектив розвитку гібридних мобільних роботів / В. А. Котенко // Автоматизація та Приладобудування («Automation and Development of Electronic Devices» ADED-2025) : збірник студентських наукових статей. – Харків : ХНУРЕ, 2025. – Вип. 2. – С. 76-80,
5. Старинець, Р. С. Моделювання руху колісного мобільного робота в закритих приміщеннях : дипломна робота ... бакалавра : 151 Автоматизація та комп'ютерно-інтегровані технології / Старинець Роман Сергійович. – Київ, 2025. – 73 с.
6. Короленко, Я. Р. Робастна система керування рухом крокуючого мобільного робота : магістерська дис. : 151 Автоматизація та комп'ютерно-інтегровані технології / Короленко Ярослав Русланович. – Київ, 2024. – 124 с.
7. Бобошко, Є. А. Моделювання мобільних роботів у програмному середовищі MATLAB : диплома робота ... бакалавра : 151 Автоматизація та комп'ютерно-інтегровані технології / Бобошко Євгеній Анатолійович. – Київ, 2024. – 76 с.

8. Привалов, О. В. Мобільний робот для автоматизації складських операцій : дипломний проєкт ... бакалавра : 151 Автоматизація та комп'ютерно-інтегровані технології / Привалов Олександр Володимирович. – Київ, 2025. – 76 с.

9. Невлюдов І. Ш. Технічне та програмне забезпечення розробки малогабаритного мобільного робота : монографія / І. Ш. Невлюдов, В. В. Євсєєв, Д. В. Гурін // Кривий Ріг: Криворізький фаховий коледж Державного некомерційного підприємства «Державний університет «Київський авіаційний інститут», 2025. – 355 с. - DOI : <https://doi.org/10.30837/978-617-8332-74-7>.

10. Jia G. W., Wang J. F. Research review of UAV swarm mission planning method. Systems Engineering and Electronics. 2021. Vol. 43. №. 1. P. 99–111. DOI: 10.3969/j.issn.1001-506X.2021.01.13

11. Канаєв В. Д. Розроблення підсистеми прийняття рішень для орієнтації робота у динамічному просторі : пояснювальна записка до кваліфікаційної роботи здобувача вищої освіти на другому (магістерському) рівні, спеціальність 151 – Автоматизація та комп'ютерно-інтегровані технології / В. Д. Канаєв ; М-во освіти і науки України, Харків. нац. ун-т радіоелектроніки. – Харків, 2024. – 58 с.

12. Кузьмук, Я. Р. Автономна навігаційна система мобільного робота: магістерська дис. : 126 Інформаційні системи та технології / Кузьмук Ярослав Романович. – Київ, 2024. – 104 с.

13. Іванюк, В. І. Система розпізнавання підводних об'єктів на базі методу Sea-thru та нейронної мережі YOLO : магістерська дис. : 123 Комп'ютерна інженерія / Іванюк Володимир Ігорович. – Київ, 2024. – 84 с.

14. Давидов Н. Д. Розроблення системи автоматизації для управління роботизованими механізмами з використанням штучного інтелекту : пояснювальна записка до атестаційної роботи здобувача вищої освіти на другому (магістерському) рівні, спеціальність 174 – Автоматизація, комп'ютерно-інтегровані технології та робототехніка / Н. Д. Давидов ; М-во освіти і науки України, Харків. нац. ун-т радіоелектроніки. – Харків, 2025. – 98 с.

15. Бездетний, Д. Д. Виявлення та захоплення швидкісних рухомих цілей глибокими нейронними мережами : магістерська дис. : 124 Системний аналіз / Бездетний Данііл Дмитрович. - Київ, 2024. - 103 с.

16. Халімонов Я. І. Забезпечення оптимальних умов на виробничих майданчиках за допомогою сенсорних технологій / Я. І. Халімонов // Комп'ютерно-інтегрованих технологій, автоматизації та робототехніки 2024 : тези доповідей І-ої Всеукр. конф., 16-17 травня 2024. – Харків, 2024. – С. 74-77.

17. ДСТУ 3400:2015. Охорона праці. Безпека при роботі з електричними установками. – К.: Національний стандарт України, 2015. – 33 с.

18. Ціль 9. Промисловість, інновації та інфраструктура // Дія, 2025. URL: https://business.diia.gov.ua/entrepreneurhandbook/item/cil_9_promislovist_innovaciya_ta_infrastruktura (дата звернення: 06.12.2025).