

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)
Розробка програмного забезпечення децентралізованого керування
групою колаборативних робіт
(тема)

Виконав:
здобувач 2 року навчання,
групи КІТІВМ-24-1
Павло ШАХОВ
(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'ютерно-інтегровані
технологічні процеси і виробництва
(повна назва освітньої програми)

Керівник доц. Світлана МАКСИМОВА
(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри КІТАР Ігор НЕВЛЮДОВ
(підпис) (власне ім'я, прізвище)

2025 р.

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

" 14 " грудня 2025 р.



Павло ШАХОВ

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'ютерно-інтегровані технологічні процеси і
виробництва
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

«15» грудня 2025 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

здобувачеві Шахову Павлу Володимировичу
(прізвище, ім'я, по батькові)

1. Тема роботи: Розробка програмного забезпечення децентралізованого
керування групою колаборативних роботів
затверджена наказом університету від 10.11.2025 р. № 1029Ст
2. Термін подання здобувачем роботи до екзаменаційної комісії 15.12.2025 р.
3. Вихідні дані до роботи 3.1 Персональний комп'ютер з процесором не нижче
Intel Core i7; 3.2 Оперативна пам'ять 16 ГБ; 3.3 Децентралізоване керування
4. Перелік питань, що потрібно опрацювати в роботі 4.1 Вступ; 4.2 Аналіз
колаборативних роботів-маніпуляторів та систем керування; 4.3 Розробка
моделі опису колаборативного робота-маніпулятора; 4.4 Розробка моделі
представлення навколишнього середовища. 4.5 Розробка моделі
децентралізованого керування групою колаборативних роботів-маніпуляторів;
4.6 Розробка метода децентралізованого керування групою колаборативних
роботів-маніпуляторів; 4.7 Вибір мови програмування та середовища розробки;
4.8 Розробка алгоритму програми для моделювання децентралізованого
керування групою колаборативних роботів-маніпуляторів; 4.9 Розробка
функцій CBF; 4.10 Розробка функцій численного моделювання;
4.11 Проведення моделювання децентралізованого керування групою
колаборативних роботів-маніпуляторів; 4.12 Аналіз отриманих результатів;
4.13 Висновки

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Демонстраційний матеріал, представлений у форматі презентації PowerPoint (*.ppt). – 14 с.ф. А 4.

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

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

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Аналіз колаборативних роботів-маніпуляторів та систем керування	02.09.2025-14.09.2025	виконано
2	Розробка моделі опису колаборативного робота-маніпулятора	15.09.2025-25.09.2025	виконано
3	Розробка моделі представлення навколишнього середовища	26.09.2025-03.10.2025	виконано
4	Розробка моделі та метода децентралізованого керування групою колаборативних роботів-маніпуляторів	04.10.2025-19.10.2025	виконано
5	Вибір мови програмування та середовища розробки	20.10.2025-05.11.2025	виконано
6	Розробка алгоритму програми для моделювання	06.11.2025-17.11.2025	виконано
7	Розробка функцій СВФ та численного моделювання	18.11.2025-29.11.2025	виконано
8	Проведення моделювання децентралізованого керування групою колаборативних роботів-маніпуляторів	30.11.2025-04.12.2025	виконано
9	Оформлення пояснювальної записки	05.12.2025-08.12.2025	виконано

Дата видачі завдання 01.09.2025 р.

Здобувач 
(підпис)

Керівник роботи _____
(підпис)

Павло ШАХОВ

(власне ім'я, прізвище)

доц. Світлана МАКСИМОВА
(посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка: 91 с., 1 табл., 8 рис., 3 дод., 21 джерело.

ДЕЦЕНТРАЛІЗОВАНЕ КЕРУВАННЯ, МУЛЬТИАГЕНТНІ СИСТЕМИ, КОЛАБОРАТИВНІ РОБОТИ, ПЛАНУВАННЯ ТРАЄКТОРІЙ, ІНДУСТРІЯ 5.0, МОДЕЛЮВАННЯ РУХУ, РОБОТОТЕХНІЧНІ СИСТЕМИ.

Мета роботи – зменшення ризику потрапляння у пастку мобільного робота, при плануванні траєкторій переміщення в рамках концепцій Індустрії 5.0.

Об’єкт дослідження – процес керування групою роботів.

Предмет дослідження – моделі та методи децентралізованого керування на базі мультиагентних систем.

В кваліфікаційній роботі проведено аналіз сучасних колаборативних роботів-маніпуляторів та систем керування ними. Проаналізовано методи групового керування та моделі мультиагентних систем. Розроблено моделі опису колаборативного робота-маніпулятора, моделі представлення навколишнього середовища та моделі децентралізованого керування групою колаборативних роботів-маніпуляторів. Розроблено метод децентралізованого керування групою колаборативних роботів-маніпуляторів. Розроблено алгоритм програми для моделювання децентралізованого керування групою колаборативних роботів-маніпуляторів, функції СВФ та функції численного моделювання. Проведено моделювання децентралізованого керування групою колаборативних роботів-маніпуляторів та аналіз отриманих результатів.

Результати, отримані при написанні кваліфікаційної 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: 91 p., 1 tables, 8 figures, 3 appendices, 21 sources.

DECENTRALIZED CONTROL, MULTIAGENT SYSTEMS, COLLABORATIVE WORK, TRAJECTORY PLANNING, INDUSTRY 5.0, MOTION MODELING, ROBOTIC SYSTEMS.

The purpose of the work is to reduce the risk of a mobile robot getting trapped when planning movement trajectories within the framework of Industry 5.0 concepts.

The object of the study is the process of controlling a group of robots.

The subject of the study is models and methods of decentralized control based on multiagent systems.

The qualification work analyzed modern collaborative robot manipulators and their control systems. Group control methods and multi-agent system models were analyzed. Collaborative robot manipulator description models, environment representation models, and decentralized control models of a group of collaborative robot manipulators were developed. A method for decentralized control of a group of collaborative robot manipulators was developed. A program algorithm was developed for modeling decentralized control of a group of collaborative robot manipulators, CBF functions, and multiple modeling functions. Decentralized control of a group of collaborative robot manipulators was simulated and the results were analyzed.

The results obtained when writing the qualification work can be attributed to Sustainable Development Goal 9 "Industry, Innovation and Infrastructure", namely 9.5. Create financial and institutional systems (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 колаборативних роботів-маніпуляторів	12
1.2 Аналіз систем керування колаборативними роботами-маніпуляторами ..	14
1.3 Аналіз методів групового керування	16
1.4 Аналіз моделі мультиагентних систем	17
1.5 Постановка задач досліджень	19
2 Розробка математичного забезпечення для моделювання децентралізованого керування групою колаборативних роботів-маніпуляторів	20
2.1 Розробка моделі опису колаборативного робота-маніпулятора.....	20
2.2 Розробка моделі представлення навколишнього середовища.....	27
2.3 Розробка моделі децентралізованого керування групою колаборативних роботів-маніпуляторів.....	31
2.4 Розробка метода децентралізованого керування групою колаборативних роботів-маніпуляторів.....	36
2.5 Висновки до 2 розділу	38
3 Розробка програми для моделювання децентралізованого керування групою колаборативних роботів-маніпуляторів.....	40
3.1 Вибір мови програмування та середовища розробки.....	40
3.2. Розробка алгоритму програми для моделювання децентралізованого керування групою колаборативних роботів-маніпуляторів	42
3.3 Розробка функцій СВФ.....	46
3.4 Розробка функцій численного моделювання	48
3.5 Висновки до 3 розділу	52
4 Проведення досліджень та аналіз отриманих результатів.....	53
4.1 Постановка задач експериментальних досліджень	53

4.2 Проведення моделювання децентралізованого керування групою колаборативних роботів-маніпуляторів.....	54
4.3 Аналіз отриманих результатів	59
4.4 Охорона праці	63
4.5 Висновки до 4 розділу	64
Висновки	66
Перелік джерел посилання	68
Додаток А Код програми численного моделювання децентралізованого керування групою колаборативних роботів-маніпуляторів у єдиній робочій зоні	72
Додаток Б Апробація результатів кваліфікаційної роботи.....	85
Додаток В Демонстраційний матеріал.....	92

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

AABB (Axis-Aligned Bounding Box / паралелепіпед) – калібрування меж вручну;

CBF – control barrier function;

EE (End-Effector) – кінцевий ефектор;

MAS (Multi-Agent Systems) – мультиагентні системи;

MPC (Model Predictive Control) – алгоритми прогнозного керування;

QP/MPC – quadratic programming / model predictive control;

SDF – signed distance field;

TM-series – techman robot.

ВСТУП

Стрімкий розвиток Індустрії 5.0 зумовлює інтеграцію колаборативних роботів у спільний робочий простір із людиною, що висуває нові вимоги до безпеки, надійності та автономності роботизованих систем. У багатьох виробничих та сервісних сценаріях роботи повинні діяти у групі, координуючи свої дії без централізованого контролера, щоб забезпечити високу гнучкість, масштабованість та відмовостійкість системи. Однією з ключових проблем при цьому є уникнення так званих «пасток» – ситуацій, коли мобільний або колаборативний робот втрачає можливість коректно продовжувати рух через локальні мінімальні точки, конфлікти в траєкторіях або обмеження середовища.

Застосування методів мультиагентних систем та децентралізованого прийняття рішень дозволяє суттєво підвищити ефективність побудови траєкторій та взаємодії між роботами, проте потребує розроблення адаптивних моделей керування, здатних забезпечити стабільність і безпечну координацію. Тому створення програмного забезпечення для моделювання та дослідження децентралізованих стратегій керування групою колаборативних роботів є актуальним завданням, яке сприяє розвитку сучасних технологій автоматизації, робототехніки та інтелектуальних виробничих систем.

Мета роботи – зменшення ризику потрапляння у пастку мобільного робота, при плануванні траєкторій переміщення в рамках концепцій Індустрії 5.0.

Об’єкт дослідження – процес керування групою роботів.

Предмет дослідження – моделі та методи децентралізованого керування на базі мультиагентних систем.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- провести аналіз сучасних колаборативних роботів-маніпуляторів;
- провести аналіз систем керування колаборативних роботів-маніпуляторів;
- провести аналіз методів групового керування;

- провести аналіз моделі мультиагентних систем;
- розробити модель опису колаборативного робота-маніпулятора;
- розробити модель представлення навколишнього середовища;
- розробити модель децентралізованого керування групою колаборативних роботів-маніпуляторів;
- розробити метод децентралізованого керування групою колаборативних роботів-маніпуляторів;
- провести вибір мови програмування та середовища розробки;
- розробити алгоритм програми для моделювання децентралізованого керування групою колаборативних роботів-маніпуляторів;
- розробити функції СВФ;
- розробити функції численного моделювання;
- провести моделювання децентралізованого керування групою колаборативних роботів-маніпуляторів;
- провести аналіз отриманих результатів.

Кваліфікаційна робота виконана згідно ДСТУ 3008–15 [1], методичних вказівок [2], з використанням статей та матеріалів із наукових публікацій, зазначених у переліку посилань, проведено апробацію на конференції [3].

1 АНАЛІЗ КОЛАБОРАТИВНИХ РОБОТІВ-МАНІПУЛЯТОРІВ ТА СИСТЕМ КЕРУВАННЯ

1.1 Аналіз сучасних колаборативних роботів-маніпуляторів

Колаборативні роботи-маніпулятори є окремим класом робототехнічних систем, розроблених для безпечної взаємодії з людиною у спільному робочому просторі. На відміну від традиційних промислових роботів, що працюють у закритих зонах та потребують відповідних систем огороження, колаборативні роботи орієнтовані на інтеграцію у гнучкі виробничі процеси, де необхідні адаптивність, мобільність та безпечне співробітництво з людиною. Сучасний ринок пропонує широкий спектр таких роботів, які відрізняються вантажопідйомністю, кількістю ступенів рухомості, рівнем автономності та інтегрованими системами безпеки, розглянемо деякі популярні компанії [4].

Серед найбільш відомих і технологічно розвинених виробників колаборативних роботів можна виділити компанії Universal Robots, FANUC, KUKA, Yaskawa, ABB, Doosan Robotics та Techman Robot. Перша лінійка роботів Universal Robots (UR3, UR5, UR10, UR16) є однією з найпоширеніших завдяки простоті програмування, компактності та інтегрованим сенсорам моменту, що забезпечують високий рівень безпеки під час взаємодії з оператором, роботи UR підтримують навчання методом демонстрації, що значно прискорює їх адаптацію до нових виробничих завдань [5].

Компанія KUKA пропонує серію LBR iiwa, орієнтовану на точні операції та роботу у спільному просторі з людиною. Роботи цієї серії оснащені високочутливими датчиками моменту у кожному ступені свободи, що дозволяє реалізувати передові алгоритми імпердансного керування та забезпечує високу чутливість до зіткнень, завдяки цьому LBR iiwa широко застосовується у задачах монтажу, лабораторній автоматизації та роботизованій інспекції [6].

Наступна лінійка – це Doosan Robotics, відома розширеним функціоналом безпеки, високою повторюваністю рухів та великою вантажопідйомністю у порівнянні з іншими колаборативними рішеннями. Роботи Doosan вирізняються інтуїтивним інтерфейсом керування та можливістю роботи в умовах, де потрібна значна точність і плавність руху [7].

Компанія FANUC пропонує серію CR та CRX, які поєднують надійність класичних промислових роботів з інтегрованими системами безпеки та функціями візуального контролю, а їх серія CRX спеціально орієнтована на простоту інтеграції та мінімальні вимоги до технічного обслуговування [8].

Окрему нішу займають роботи Techman Robot (TM-series), що мають вбудовані системи машинного зору, інтегровані у конструкцію маніпулятора, що дозволяє виконувати складні задачі розпізнавання та позиціонування без зовнішніх камер, що знижує загальну вартість системи та підвищує гнучкість її застосування [9].

Аналізуючи вище перераховані сучасні колаборативні роботи-маніпулятори, можна виділити такі основні переваги:

- високий рівень безпеки, що забезпечується сенсорами моменту, контролем сил взаємодії, обмеженням швидкості та автоматичним реагуванням на зіткнення;
- простота навчання та програмування, зокрема завдяки графічним інтерфейсам та навчанням шляхом демонстрації;
- гнучкість інтеграції у виробничі процеси без необхідності складних огорожувальних конструкцій;
- компактність та малий інерційний момент, що дозволяє використовувати їх у тісних робочих зонах;
- зменшене енергоспоживання порівняно з традиційними промисловими маніпуляторами;
- висока адаптивність, що забезпечується використанням алгоритмів імпедансного керування, систем технічного зору та методів кооперації.

Особливістю сучасних колаборативних роботів є поєднання високої точності рухів із розвиненими механізмами взаємодії з людиною, що дозволяє використовувати їх у задачах монтажу, пакування, контролю якості, лабораторних дослідженнях та сервісній робототехніці. Здатність працювати в команді з іншими роботами та людиною робить їх ключовим елементом концепції Індустрії 5.0, орієнтованої на людинороботну колаборацію та інтелектуальні виробничі системи [10].

1.2 Аналіз систем керування колаборативними роботами-маніпуляторами

Системи керування колаборативних роботів-маніпуляторів мають специфічні вимоги, пов'язані з необхідністю безпечної взаємодії робота з людиною та іншими роботами у спільному робочому просторі. На відміну від класичних промислових систем керування, орієнтованих на жорстку автоматизацію та максимальну швидкодію, системи керування колаборативних роботів характеризуються підвищеною гнучкістю, адаптивністю та використанням сенсорної інформації для оперативної зміни поведінки робота. Це зумовлює застосування нових архітектур керування, інтелектуальних алгоритмів та інтеграцію мультимодальних систем сприйняття. Більшість сучасних колаборативних роботів використовують модульну ієрархічну структуру керування, яка включає рівні низького, середнього та високого керування [11].

На нижньому рівні реалізуються задачі позиційного, швидкісного та силового регулювання, що забезпечуються приводами із зворотним зв'язком, високоточними енкодерами та датчиками моменту. Для багатьох robots, таких як KUKA LBR iiwa, Doosan або Universal Robots, характерним є застосування імпедансного та адмітантного керування, що дозволяє реалізувати плавну взаємодію з зовнішніми об'єктами та компенсувати непередбачувані навантаження.

На середньому рівні здійснюється формування рухових траєкторій, контроль обмежень безпеки та перевірка зіткнень. У цьому контексті широко застосовуються алгоритми прогнозного керування (Model Predictive Control), контролери обмеженої взаємодії та стратегія управління з обмеженням сили контакту. Для безпечної роботи з людиною активно використовуються функції обмеження швидкості, зони безпеки та адаптивні обмеження моменту, що відповідають стандартам ISO/TS 15066.

На високому рівні системи керування колаборативних роботів використовують інтелектуальні методи прийняття рішень: моделі машинного зору, навчання з підкріпленням, планування рухів у складному середовищі та аналіз оточення в режимі реального часу. Завдяки цим технологіям робот може адаптувати поведінку до поточної ситуації, координувати дії з іншими агентами та виконувати завдання більш автономно. Важливу роль відіграє підтримка навчання шляхом демонстрації, де оператор задає траєкторію вручну або через сенсорні пристрої, а система керування автоматично генерує відповідну поведінкову модель. Окремою складовою сучасних систем керування є засоби забезпечення безпеки, що інтегрують сенсорні системи:

- датчики сили та моменту в суглобах;
- 3D-візуальні системи контролю робочої зони;
- системи моніторингу швидкості та положення;
- технології прогнозування колізій.

Завдяки цьому робот може визначати присутність оператора, адаптувати свою швидкість та динамічні параметри, а також запобігати небезпечним ситуаціям під час спільної роботи [12].

У сучасних виробничих системах все більшої популярності набувають децентралізовані та мережеві архітектури керування, які дозволяють групі колаборативних роботів узгоджувати свої дії без необхідності центрального контролера. Це дає змогу підвищити масштабованість, відмовостійкість системи та ефективність спільного виконання завдань, особливо в умовах невизначеності та змінних зовнішніх впливів. У таких системах широко застосовуються підходи,

основані на мультиагентних моделях, включно з алгоритмами координації, розподіленим плануванням і поведінковими моделями.

Таким чином, системи керування колаборативних роботів-маніпуляторів характеризуються комплексною структурою та широким спектром алгоритмічних підходів, спрямованих на забезпечення безпеки, високої адаптивності та ефективної взаємодії з людиною й іншими роботами. Розвиток децентралізованих моделей керування та інтелектуальних алгоритмів координації робить ці системи ключовим елементом робототехніки нового покоління в рамках концепцій Індустрії 5.0 [13].

1.3 Аналіз методів групового керування

Групове або кооперативне керування роботами є важливим аспектом створення багатороботних систем, здатних автономно виконувати спільні задачі у складному, динамічному середовищі. Сучасні методи можна поділити на чотири основні категорії, такі як централізовані, децентралізовані, ієрархічні та методи на основі поведінкових моделей [14].

Першим важливим підходом є централізоване керування, головною перевагою якого виступає високий рівень глобальної оптимізації: система отримує повну інформацію про стан усіх роботів і середовища, що дозволяє генерувати ефективні та узгоджені траєкторії. Водночас цей підхід має суттєвий недолік – залежність від одного центру прийняття рішень, що робить систему вразливою до відмови та створює значне навантаження на канали зв'язку, ускладнюючи масштабування на більші групи роботів.

На протизагу централізованим методам, децентралізовані схеми демонструють високу стійкість і автономність: кожен робот може працювати навіть за відсутності постійного зв'язку з іншими, ухвалюючи рішення локально, такий підхід добре масштабується та легко адаптується до змін у середовищі. Однак ці переваги поєднуються з недоліками локальності – роботи не завжди

можуть оцінити глобальну картину, що інколи призводить до субоптимальних траєкторій, локальних мінімумів або конфліктів руху [15].

Методи на основі поведінкових моделей дають змогу реалізувати природну для колективних систем динаміку: роботи формують групову поведінку через прості правила, такі як вирівнювання, уникнення зіткнень і притягання до центру групи. Перевагою є простота реалізації та швидкість реакції на зміни. Проте така простота породжує і недоліки: поведінкові алгоритми можуть бути недостатньо точними для складних виробничих сценаріїв і схильні до нестабільних станів за наявності зовнішніх збурень.

Окрему категорію становлять методи на основі оптимізації та навчання, які забезпечують найвищу адаптивність і здатність роботи покращувати свою координацію з часом. Їхньою перевагою є можливість генерувати оптимальні стратегії навіть у складних, динамічних середовищах. Проте ці методи мають і серйозні обмеження – високу обчислювальну складність, потребу у великих обсягах тренувальних даних та складність гарантування безпеки в режимі реального часу, що обмежує їх застосування в критично важливих робототехнічних системах [16].

1.4 Аналіз моделі мультиагентних систем

Мультиагентні системи (MAS, Multi-Agent Systems) є основою сучасних підходів до децентралізованого керування групою роботів. Вони представляють собою сукупність автономних агентів, що взаємодіють між собою та з навколишнім середовищем для досягнення індивідуальних або колективних цілей. Кожний агент у MAS має власні сенсори, приводи та алгоритми прийняття рішень, що дозволяє йому оцінювати стан середовища, обмінюватися інформацією з іншими агентами та ухвалювати локальні рішення.

Основна перевага мультиагентного підходу полягає у децентралізації управління, що підвищує масштабованість та відмовостійкість системи. Кожен агент працює автономно, а координація групи здійснюється через обмін

локальною інформацією, правила взаємодії або глобальні сигнали, що дозволяє уникнути єдиної точки відмови, характерної для централізованих систем, та підвищує стійкість до непередбачуваних змін у середовищі [17].

Аналіз існуючих моделей MAS показує, що вони включають кілька ключових компонентів, по-перше, це самі агенти, які є автономними роботизованими вузлами з можливістю сприйняття, обробки інформації та виконання дій у середовищі. По-друге, середовище виступає як модель робочого простору, враховуючи перешкоди, динаміку об'єктів та інших агентів, що створює основу для побудови траєкторій і планування дій. По-третє, комунікаційна структура визначає, як агенти обмінюються даними; вона може бути повною, частковою або обмеженою топологією мережі, що безпосередньо впливає на швидкість координації та надійність системи. Правила взаємодії та координації визначають, яким чином агенти ухвалюють рішення про рух, розподіл завдань, уникнення конфліктів і досягнення спільних цілей.

Сучасні реалізації MAS у робототехніці відрізняються різними підходами до поведінки агентів. Кооперативні системи передбачають активний обмін інформацією між агентами та спільну роботу для досягнення колективної мети. Конкурентні системи забезпечують автономну діяльність агентів, де взаємодія обмежується правилами уникнення конфліктів або перешкод. Гібридні підходи поєднують кооперативну та автономну поведінку залежно від завдань і контексту роботи, що дозволяє підвищити ефективність у складних сценаріях з великою кількістю агентів [18].

До основних переваг мультиагентних систем у керуванні групою роботів належать висока відмовостійкість завдяки децентралізованій архітектурі, масштабованість, адаптивність до змін середовища та можливість гнучкого розподілу ролей між агентами. Але існують і такі обмеження як підвищена складність розробки та тестування алгоритмів координації, залежність від достовірності локальних сенсорних даних, ризик виникнення непередбачуваних поведінкових патернів у великих групах та складність забезпечення глобальної оптимальності при децентралізованому керуванні.

Тобто, можна зробити висновок, що моделі мультиагентних систем забезпечують потужну платформу для реалізації децентралізованого керування групою колаборативних роботів, вони поєднують автономність, кооперацію та адаптивність агентів, що робить їх ключовими для побудови ефективних, безпечних та гнучких робототехнічних систем у рамках концепцій Індустрії 5.0 [19].

1.5 Постановка задач досліджень

На основі проведеного аналізу сучасних колаборативних роботів-маніпуляторів, систем їх керування, методів групового керування та моделей мультиагентних систем визначено основні проблеми, що впливають на ефективність децентралізованої взаємодії роботів у спільному робочому просторі. Актуальною є проблема уникнення «пасток» та локальних мінімумів під час планування траєкторій, а також забезпечення безпечної та узгодженої поведінки роботів-агентів без централізованого контролера. Відповідно до поставленої мети кваліфікаційної роботи необхідно виконати такі завдання:

- розробити модель опису колаборативного робота-маніпулятора;
- розробити модель представлення навколишнього середовища;
- розробити модель децентралізованого керування групою колаборативних роботів-маніпуляторів;
- розробити метод децентралізованого керування групою колаборативних роботів-маніпуляторів;
- розробити алгоритм програми для моделювання децентралізованого керування групою колаборативних роботів-маніпуляторів;
- розробити функції CBF;
- розробити функції численного моделювання;
- провести моделювання децентралізованого керування групою колаборативних роботів-маніпуляторів;
- провести аналіз отриманих результатів.

2 РОЗРОБКА МАТЕМАТИЧНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ МОДЕЛЮВАННЯ ДЕЦЕНТРАЛІЗОВАНОГО КЕРУВАННЯ ГРУПОЮ КОЛАБОРАТИВНИХ РОБОТІВ-МАНІПУЛЯТОРІВ

2.1 Розробка моделі опису колаборативного робота-маніпулятора

Розробка математичної моделі опису колаборативного робота-маніпулятора є критичною умовою коректного моделювання децентралізованого керування у спільній зоні з людиною, оскільки без адекватного формалізму неможливо узгодити локальні цілі агентів із глобальними вимогами безпеки та продуктивності. Кінематична модель забезпечує однозначне відображення між простором суглобів і простором задач, дозволяючи формулювати локальні мікрозадачі у вигляді положення та орієнтації інструмента, узгоджувати їх із траєкторіями сусідів і з проксемічними обмеженнями людини. Вона є основою для обчислення Якобіана, маніпулювальності та для проєкції колективних критеріїв у швидкісні команди, що безпосередньо потрібні у децентралізованих QR/MPC. Динамічна модель, своєю чергою, дозволяє врахувати інерційні, коріолісові та гравітаційні ефекти, без чого неможливо гарантувати стійкість імпедансної взаємодії, енергетичну узгодженість та коректну реакцію на контактні сили людини. Саме динаміка надає змогу встановити фізично обґрунтовані межі моментів, швидкостей і прискорень, що робить реальними бар'єрні умови безпеки та їх виконання у реальному часі. У мультиагентній постановці ці моделі потрібні для консенсусу намірів: кінематика задає геометрію конфліктів, динаміка – їх часову і силову реалізованість. Вони також є базою для ідентифікації параметрів і адаптації, що знижує невизначеність під час спільної роботи з людиною і в мінливому середовищі. Без чіткої кінематики неможливо коректно визначити відстані до перешкод і людини, а без динаміки – оцінити ризики перенавантаження при ухиленні або кооперативному хваті. Обидві моделі забезпечують формування єдиних критеріїв вартості та обмежень,

придатних до децентралізованого розв'язання з локальним обміном ескізами траєкторій. Таким чином, кінематична та динамічна компоненти разом створюють необхідну та достатню основу для безпечної, узгодженої та ефективної координації групи колаборативних маніпуляторів поруч із людиною, і подальший виклад природно починається з побудови кінематичної моделі колаборативного робота-маніпулятора [3].

Припустимо що, узагальнені координати $q_i \in \mathbb{R}^{n_i}$, які використовуються для опису положення та орієнтації всіх рухомих ланок i -го робота-маніпулятора у просторі суглобів. Їх основне призначення – забезпечити компактне математичне представлення конфігурації робота в будь-який момент часу, яке однозначно визначає його геометричний стан у просторі:

$$q_i \in \mathbb{R}^{n_i}, \quad (2.1)$$

де q_i – вектор, елементи якого відповідають незалежним ступеням свободи маніпулятора, наприклад, кутам повороту обертових з'єднань або лінійним переміщенням призматичних;

n_i – кількість ступенів свободи i -го робота (наприклад, для 6-осьового маніпулятора $n_i = 6$);

\dot{q}_i – швидкість;

\ddot{q}_i – швидкості та прискорення.

Ці координати служать зв'язною ланкою між локальною динамікою робота та глобальною мультиагентною моделлю, забезпечуючи єдину систему опису для оптимізації рухів, контролю обмежень і виконання задач у спільному робочому просторі.

Для опису залежності між узагальненими координатами маніпулятора q_i (тобто конфігурацією суглобів) та положенням і орієнтацією його кінцевого ефектора (End-Effector, EE) у робочому просторі під позою EE розуміють сукупність положення (позиції) та орієнтації інструмента або захватного пристрою робота у тривимірному просторі. Іншими словами, це повний опис

того, де знаходиться ефектор і як він орієнтований відносно світової системи координат. У дослідженні децентралізованого керування групою колаборативних роботів-маніпуляторів у єдиній робочій зоні з людиною модель прямої кінематики є ключовою для:

- обчислення реального положення ЕЕ на основі стану суглобів, щоб оцінити можливі колізії між роботами та людиною;
- формування локальних та глобальних цілей (наприклад, досягнення заданої точки в просторі, збереження безпечної відстані від людини);
- реалізації зворотних задач керування, де на основі бажаної пози/орієнтації потрібно знаходити відповідні узагальнені координати;
- синхронізації роботи кількох роботів, прямокінематичне представлення дозволяє швидко визначати взаємне розташування ефекторів у робочій зоні. Математична модель прямої кінематики буде мати наступний вигляд:

$$x_i = f_i(q_i) \in SE(3), \quad (2.2)$$

де x_i – поза кінцевого ефектора i -го робота в робочому просторі, що включає позицію та орієнтацію;

$f_i(q_i)$ – пряма кінематична функція, яка обчислює позу ЕЕ на основі узагальнених координат q_i ;

$SE(3)$ – спеціальна евклідова група жорстких перетворень у 3D, яка включає обертання та перенесення. Вона описує всі можливі пози та орієнтації в тривимірному просторі.

Проведемо декомпозицію моделі пози та орієнтації (2.2):

$$x_i = \{p_i, R_i\}, \quad (2.3)$$

де $p_i \in \mathbb{R}^3$ – вектор позиції кінцевого ефектора у глобальній (або локальній) системі координат, його призначення визначає, де знаходиться робочий інструмент робота, та складається з наступних параметрів:

$$p_i = [x, y, z]^T, \quad (2.4)$$

де x – координата вздовж осі X робочого простору та визначає, наскільки далеко вліво або вправо розташований ефектор. Використовується для обчислення горизонтальних переміщень;

y – координата вздовж осі Y та визначає наскільки ефектор зміщений у глибину або назовні (перпендикулярно до осі Y). Використовується для побудови траєкторій у площині XY ;

z – координата вздовж осі Z та визначає висоту ефектора відносно базової площини. Є ключова для маніпуляцій над об'єктами (піднімання, опускання).

$R_i \in SO(3)$ – матриця обертання (3×3), яка описує орієнтацію кінцевого ефектора відносно глобальної системи координат та визначає, як саме орієнтований захват або інструмент (наприклад, кут нахилу при взаємодії з деталлю або уникнення зіткнення). Належить до спеціальної ортогональної групи $SO(3)$, що містить всі матриці повороту у 3D та використовується у маніпуляційних завданнях, де важлива не тільки позиція, а й орієнтація (наприклад, складання деталей, спільна робота з людиною).

Математична модель прямої кінематики (2.2), є базою для побудови кінематичних ланцюгів, обчислення Якобіана для керування швидкістю та силою, а також для обміну інформацією між агентами у децентралізованій архітектурі. Вона дозволяє узгодити локальні задачі робота (його конфігурацію) з глобальними цілями системи (розташування у спільному просторі, безпечна взаємодія з людиною).

Використання обчислення Якобіана в даному дослідженні є необхідним для опису зв'язку між узагальненими координатами робота-маніпулятора та його швидкісними характеристиками в робочому просторі. Якобіан дозволяє трансформувати швидкості та сили з простору зчленувань у декартовий простір кінцевого ефектора, що є критично важливим для реалізації децентралізованого керування та координації руху групи роботів. Він забезпечує можливість розрахунку траєкторій, уникнення зіткнень, а також врахування обмежень

динаміки при взаємодії з людиною в єдиній робочій зоні. Завдяки Якобіану стає можливим реалізувати точне керування положенням і орієнтацією ефектора при зміні узагальнених координат у реальному часі:

$$\dot{x}_i = J_i(\dot{q}_i), \text{ де } J_i \in \mathbb{R}^{6 \times n_i}, \quad (2.5)$$

де \dot{x}_i – вектор швидкості кінцевого ефектора (ЕЕ) у просторі робочих координат, який складається з лінійної та кутової швидкостей. Його розмірність дорівнює 6 (3 для поступального руху та 3 для обертального);

J_i – матриця Якобіана, яка є функцією узагальнених координат q_i і відображає кінематичний зв'язок між простором зчленувань та простором робочих координат. Розмірність J_i дорівнює $6 \times n_i$, де 6 відповідає кількості степенів свободи в робочому просторі, а n_i – кількості степенів свободи маніпулятора;

\dot{q}_i – вектор швидкостей узагальнених координат (суглобових кутів або зсувів), який визначає, як швидко змінюється конфігурація робота.

У рамках даного дослідження, модель (2.5) використовується для забезпечення точного керування траєкторією руху ЕЕ при зміні суглобових параметрів, для синхронізації рухів групи маніпуляторів у децентралізованій системі, а також для реалізації стратегій уникнення зіткнень з людиною та іншими роботами в робочій зоні.

Введемо обмеження по суглобам, це обґрунтовано тим, що реальні маніпулятори мають конструктивні та безпекові обмеження, які не дозволяють безмежне обертання або переміщення ланок. Кожен суглоб має допустимий діапазон кутів або лінійних переміщень, обумовлений механічною конструкцією, а також має обмеження швидкості для запобігання надмірним інерційним навантаженням, що можуть призвести до втрати точності або пошкодження. У контексті даного дослідження введення таких обмежень гарантує безпечну взаємодію колаборативних роботів між собою та з людиною, що особливо важливо у спільному робочому просторі:

$$q_i^{min} \leq q_i \leq q_i^{max}, \quad |\dot{q}_i| \leq \dot{q}_i^{max}, \quad (2.6)$$

де q_i^{min} – мінімально допустиме значення узагальненої координати для i -го суглоба (нижня межа кута або переміщення);

q_i^{max} – максимально допустиме значення узагальненої координати для i -го суглоба (верхня межа кута або переміщення);

q_i – поточне значення узагальненої координати для i -го суглоба, яке повинно залишатися в межах фізично допустимого діапазону;

$|\dot{q}_i|$ – модуль швидкості зміни узагальненої координати, що визначає реальну динаміку руху суглоба в даний момент;

\dot{q}_i^{max} – максимально допустима швидкість для i -го суглоба, яка обмежує динамічні навантаження та гарантує плавність рухів.

Такі обмеження дозволяють формувати керуючі сигнали, що враховують фізичні характеристики роботів, підвищують надійність керування та запобігають аварійним ситуаціям при децентралізованій взаємодії кількох маніпуляторів у спільній робочій зоні [3].

Для опису динаміки колаборативного робота-маніпулятора, пропонується використовувати моделі Лагранжа-Ейлера, обґрунтований тим, що вона забезпечує узагальнений підхід до врахування кінематичних та динамічних характеристик системи з багатьма ступенями вільності. Ця модель дозволяє описати рух за допомогою узагальнених координат, що є особливо важливим у мультиагентному середовищі, де роботи взаємодіють у спільному робочому просторі. Метод Лагранжа забезпечує чіткий зв'язок між енергією системи (кінетичною та потенціальною) і прикладеними узагальненими силами, що дає можливість врахувати не лише жорсткі з'єднання, але й обмеження по суглобах та зовнішні взаємодії. Крім того, він є зручним для побудови адаптивних алгоритмів керування, оскільки дозволяє легко вводити додаткові члени для компенсації тертя, гравітації або сил зіткнення з людиною. Застосування цієї моделі у децентралізованому керуванні дає можливість формувати локальні закони керування з урахуванням динамічних параметрів, що підвищує безпеку і

точність спільних дій. У результаті динаміка робота описується рівнянням Лагранжа другого роду у формі:

$$M_i(q_i)\ddot{q}_i + C_i(q_i, \dot{q}_i) + g_i(q_i) + f_i^{fr}(\dot{q}_i) = \tau_i + J_i^T(q_i)F_i^{ext}, \quad (2.7)$$

де $M_i(q_i)$ – матриця інерції (розмірність $n_i \times n_i$), яка відображає залежність узагальнених мас від положення суглобів; визначає, як система реагує на прискорення;

\ddot{q}_i – вектор узагальнених прискорень, що описує динамічний стан суглобів;

$C_i(q_i, \dot{q}_i)$ – вектор коріолісових та відцентрових сил, що виникають при русі; він моделює інерційні ефекти, пов'язані зі швидкістю;

$g_i(q_i)$ – вектор сил гравітації, що діє на ланки маніпулятора залежно від положення в просторі;

$f_i^{fr}(\dot{q}_i)$ – сили та моменти тертя у суглобах, які залежать від швидкості та враховують втрати енергії;

τ_i – вектор керуючих моментів, що подаються на приводи для реалізації руху;

$J_i^T(q_i)$ – транспонована матриця Якобіана, що відображає зв'язок між просторовими та суглобовими силами. Наприклад, якщо розглядати 6-осьовий маніпулятор з узагальненими координатами $q_i = [q_{i1}, q_{i2}, \dots, q_{i6}]^T$, а ефектор у 3D-просторі піддається зовнішній силі $F_i^{ext} = [F_x, F_y, F_z, M_x, M_y, M_z]^T$, то транспонована матриця Якобіана має розмірність 6×6 і може мати вигляд:

$$J_i^T(q_i) = \begin{bmatrix} \frac{\partial x}{\partial q_{i1}} & \frac{\partial y}{\partial q_{i1}} & \frac{\partial z}{\partial q_{i1}} & \frac{\partial \phi_x}{\partial q_{i1}} & \frac{\partial \phi_y}{\partial q_{i1}} & \frac{\partial \phi_z}{\partial q_{i1}} \\ \frac{\partial x}{\partial q_{i2}} & \frac{\partial y}{\partial q_{i2}} & \frac{\partial z}{\partial q_{i2}} & \frac{\partial \phi_x}{\partial q_{i2}} & \frac{\partial \phi_y}{\partial q_{i2}} & \frac{\partial \phi_z}{\partial q_{i2}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial x}{\partial q_{i6}} & \frac{\partial y}{\partial q_{i6}} & \frac{\partial z}{\partial q_{i6}} & \frac{\partial \phi_x}{\partial q_{i6}} & \frac{\partial \phi_y}{\partial q_{i6}} & \frac{\partial \phi_z}{\partial q_{i6}} \end{bmatrix}, \quad (2.8)$$

де (x, y, z) – координати позиції ефектора;

ϕ_x, ϕ_y, ϕ_z – параметри орієнтації (наприклад, кути Ейлера) залежно від суглобових координат q_i .

При застосуванні у динамічній моделі (2.7), добуток $J_i^T(q_i)F_i^{ext}$ перетворює зовнішню силу та момент у відповідні моменти на приводах суглобів, що дозволяє враховувати взаємодію робота з людиною або іншими агентами у спільній робочій зоні;

F_i^{ext} – вектор зовнішніх сил та моментів, що діють на ефектор (наприклад, при взаємодії з людиною або іншим роботом) [3].

Модель (2.7) дозволяє врахувати внутрішні динамічні ефекти та зовнішні впливи, що є критично важливим у дослідженні децентралізованого керування в мультиагентній системі, де присутні взаємодії з іншими роботами та оператором.

2.2 Розробка моделі представлення навколишнього середовища

Розробка моделі представлення навколишнього середовища є ключовим етапом для забезпечення безпечної та ефективної роботи колаборативних роботів у виробничих умовах. У динамічних середовищах, де присутні люди, рухомі об'єкти та інші роботи, необхідно створювати математичні структури, що дозволяють точно описувати просторові характеристики робочої зони. Такі моделі мають враховувати не лише геометрію, а й змінність положення об'єктів у часі, що забезпечує можливість прогнозування та уникнення зіткнень. Для цього необхідне використання методів просторової дискретизації та адаптивного оновлення інформації на основі сенсорних даних. Результатом даного етапу є побудова узагальненої моделі робочого простору з інтегрованим описом перешкод, яка стане базою для алгоритмів планування траєкторії та адаптивного керування рухом маніпулятора.

На першому кроці необхідно провести дослідження узгодження всіх геометричних величин (позиції, орієнтації, відстані) для планування й перевірки безпеки. Нехай кожен агент має базову рамку $\{B_i\}$, глобальну/сценову рамку $\{W\}$ та рамки сенсорів $\{S\}$. Калібрування дає перетворення:

$$\mathcal{F} \{W_{T_{B_i}}, W_{T_s}\} \in SE(3), \quad (2.9)$$

де $W_{T_{B_i}}$ – гомогенне перетворення бази робота i у світову рамку та задає де «стоїть» робот;

W_{T_s} – пози/орієнтації сенсорів та потрібні для проекції вимірів у карту;

$\mathcal{F} = \{W, B_i, S\}$ – набір рамок для перетворень і об'єднання даних.

Представимо робочу зону як: $\mathcal{W} \subset \mathbb{R}^3$. Отримання даних про робочу зону (карту місцевості) пропишемо за наступним принципом:

- креслення ділянки \rightarrow імпорт габаритів;
- SLAM/сканування LiDAR/RGB-D \rightarrow груба карта;
- калібрування меж вручну (Axis-Aligned Bounding Box (AABB)/паралелепіпед) або як поліедр. AABB – це вісьово-вирівняний обмежувальний паралелепіпед (прямокутний паралелепіпед), сторони якого паралельні осям глобальної системи координат. Використання AABB дозволяє ефективно описати перешкоди в робочій зоні та спростити алгоритми зіткнень при побудові траєкторій мобільного або маніпуляційного робота:

$$\mathcal{W} = \{x \in \mathbb{R}^3: Hx \leq h\}, \quad (2.10)$$

де \mathcal{W} – множина всіх допустимих точок робочого простору робота (область, в якій він може виконувати операції);

$x \in \mathbb{R}^3$ – вектор координат точки у тривимірному просторі, що визначає положення певної точки робочої зони ($x = [x, y, z]^T$);

H – матриця розмірності $m \times 3$, що містить коефіцієнти для лінійних обмежень, де кожен рядок описує одну площину, яка обмежує робочу зону;

h – вектор розмірності $m \times 1$, що задає граничні значення для нерівностей, разом з H формує межі робочого простору;

$H_x \leq h$ – система лінійних нерівностей, яка описує політоп (багатогранник), що визначає робочу зону робота (наприклад, паралелепіпед або довільну опуклу область).

Або як бокс (AABB):

$$[x_{min}, x_{max}] \cdot [y_{min}, y_{max}] \cdot [z_{min}, z_{max}], \quad (2.11)$$

де $[x_{min}, x_{max}]$ – межі по осі X , що визначають найменше та найбільше допустиме значення координати X для об'єкта (робочої зони або перешкоди);

$[y_{min}, y_{max}]$ – межі по осі Y , що обмежують простір по ширині;

$[z_{min}, z_{max}]$ – межі по осі Z , що обмежують висоту області.

Математичне представлення (2.11) дозволяє математично формалізувати межі робочої області, щоб уникати виходу робота за допустимі межі та враховувати геометрію простору при плануванні траєкторій та уникненні зіткнень. А вираз (2.11) – бокс описує об'ємну область у 3D-просторі, в якій робот може рухатись або яку займає перешкода.

Для моделювання перешкод у робочій зоні колаборативного робота ($\mathcal{O} = \{\mathcal{O}_k\}$) використовується опис у вигляді множини обмежень у тривимірному просторі, що задають об'єм зайнятий об'єктом. Найчастіше перешкоди представляються у вигляді AABB (Axis-Aligned Bounding Box) або паралелепіпеда:

$$\mathcal{O}_k = \{x \in \mathbb{R}^3: x_{min}^k \leq x \leq x_{max}^k, y_{min}^k \leq y \leq y_{max}^k, z_{min}^k \leq z \leq z_{max}^k\}, \quad (2.12)$$

де \mathcal{O}_k – множина точок, що належать перешкоді k – го об'єкта;

x_{min}^k, x_{max}^k – мінімальна та максимальна координати перешкоди по осі X , визначають довжину об'єкта;

y_{min}^k, y_{max}^k – мінімальна та максимальна координати по осі Y , визначають ширину об'єкта;

z_{min}^k, z_{max}^k – межі по осі Z , що визначають висоту перешкоди;

k – індекс перешкоди (для випадку, коли є кілька перешкод у робочій зоні);
 \mathbb{R}^3 – простір, у якому описуються координати об'єктів (тривимірний робочий простір).

Для визначення, чи знаходиться точка (наприклад, позиція робота) всередині перешкоди, пропонується вести функцію індикатора:

$$\chi_{\mathcal{O}_k}(x) = \begin{cases} 1, & \text{якщо } x \in \mathcal{O}_k \\ 0, & \text{якщо } x \notin \mathcal{O}_k \end{cases} \quad (2.13)$$

Для опису руху центру перешкоди вводиться траєкторія:

$$p_k(t) = [x_k(t), y_k(t), z_k(t)]^T, \quad (2.14)$$

де $p_k(t)$ – положення центру перешкоди.

Швидкість та прискорення динамічної перешкоди:

$$\dot{p}_k(t) = \frac{d}{dt} p_k(t), \quad \ddot{p}_k(t) = \frac{d^2}{dt^2} p_k(t), \quad (2.15)$$

де $\dot{p}_k(t)$ – перша похідна від $p_k(t)$ за часом, тобто швидкість руху перешкоди. Вона показує, як швидко змінюється положення об'єкта в кожній з координат x, y, z у часі;

$\ddot{p}_k(t)$ – друга похідна від $p_k(t)$ за часом, тобто прискорення руху перешкоди. Воно характеризує, наскільки швидко змінюється швидкість об'єкта (чи він пришвидшується, чи сповільнюється) у просторі.

Розробимо умову уникнення зіткнення для робота з динамічною перешкодою:

$$\|p_r(t) - p_k(t)\| \geq d_{safe}, \quad \forall t \in [0, T], \quad (2.16)$$

де $p_r(t)$ – траєкторія робота;

$p_k(t)$ – траєкторія перешкоди;

d_{safe} – мінімальна безпечна відстань;

T – горизонт планування.

2.3 Розробка моделі децентралізованого керування групою колаборативних роботів-маніпуляторів

Модель децентралізованого керування групою роботів обрана для забезпечення масштабованості та відмовостійкості системи, оскільки відсутність єдиного центру керування мінімізує ризик збоїв при виході з ладу окремого вузла. Такий підхід дозволяє кожному роботу приймати рішення на основі локальної інформації та обміну даними з сусідніми агентами, що знижує затримки і покращує адаптивність у динамічному середовищі. Для формалізації взаємодії між агентами використовується комунікаційний граф, який відображає структуру зв'язків між вузлами та гарантує узгодженість дій через алгоритми консенсусу. Комунікаційний граф дає можливість визначити, які роботи обмінюються даними, і врахувати обмеження на дальність зв'язку. У рамках дослідження комунікаційний граф описується як орієнтований або неорієнтований граф, що забезпечує симетрію або асиметрію обміну даними:

$$\begin{aligned} \mathcal{G}(t) &= (\mathcal{V}, \mathcal{E}(t)), |\mathcal{V}| = N, \\ \mathcal{N}_i(t) &= \{j: (i, j) \in \mathcal{E}(t)\}, \end{aligned} \tag{2.17}$$

де \mathcal{V} – множину вершин, де кожна вершина відповідає окремому роботу або агенту системи;

$\mathcal{E}(t)$ – множина орієнтованих або неорієнтованих ребер у момент часу t , які визначають наявність активних каналів зв'язку між роботами;

$|\mathcal{V}| = N$ – загальна кількість роботів (вузлів) у системі, що визначає розмір мережі;

$\mathcal{N}_i(t)$ – множина сусідів вузла i у момент часу t , тобто тих агентів, з якими робот i може безпосередньо обмінюватися даними.

В загальні $\mathcal{N}_i(t) = \{j: (i, j) \in \mathcal{E}(t)\}$ вказує на те, що зв'язок існує, якщо ребро з i на j належить множині $\mathcal{E}(t)$. Таким чином, ці параметри дозволяють динамічно описати топологію зв'язків у групі роботів і забезпечити коректну роботу децентралізованих алгоритмів керування.

Бар'єрні умови безпеки Control Barrier Functions (CBF) у просторі кінцевого ефектора (ЕЕ) представимо трьома випадками:

– міжроботні відстані (запобігає зіткненням роботів, забезпечуючи мінімальну безпечну відстань):

$$h_{ij}(p_i, p_j) = \|p_i - p_j\|^2 - (r_i + r_j + d_m)^2 \geq 0; \quad (2.18)$$

– від людини (визначає еліптичну зону безпеки для людини, враховуючи її прогнозований рух і невизначеність):

$$h_{iH}(p_i, p_H) = (p_i - p_H)^T \Sigma_H^{-1}(t)(p_i - p_H) - R_{safe}^2(t) \geq 0; \quad (2.19)$$

– від перешкод через SDF (умова гарантує уникнення зіткнень з об'єктами або стінами, враховуючи змінне оточення):

$$h_{iO}(p_i) = \phi(p_i, t) - d_{min} \geq 0, \quad (2.20)$$

де p_i, p_j – позиції роботів i та j у просторі;

$\|p_i - p_j\|^2$ – квадрат відстані між двома роботами;

r_i, r_j – радіуси безпечних зон для кожного робота (враховують їхні габарити);

d_m – додатковий мінімальний буфер безпеки між роботами;

p_H – позиція людини в робочій зоні;

$R_{safe}^2(t)$ – динамічний радіус безпечної зони навколо людини;

$\phi(p_i, t)$ – відстань від робота до найближчої статичної або динамічної перешкоди в момент часу t ;

d_{min} – мінімально допустима відстань до перешкоди.

Модель людини представимо відповідно як динамічну перешкоду, та складається з наступних чотирьох базових параметрів:

– стан людини:

$$\psi_H = [p_H^T, \dot{p}_H^T]^T \in \mathbb{R}^6, \quad (2.21)$$

де ψ_H – вектор стану людини у робочому просторі робота, що включає як позицію, так і швидкість;

$p_H \in \mathbb{R}^3$ – вектор просторових координат (позиція) людини у тривимірному просторі, зазвичай у системі координат робочої зони або базової системи робота;

$\dot{p}_H \in \mathbb{R}^3$ – вектор лінійних швидкостей руху людини (зміна позиції в часі);

p_H^T – транспонований вектор позиції, використовується для формування повного стану в єдину колонку;

\dot{p}_H^T – транспонований вектор швидкостей;

\mathbb{R}^6 – розмірність простору стану: 3 координати для позиції та 3 для швидкості, разом 6 компонент.

– прогноз траєкторії на горизонт прогнозування в часі (T_H):

$$T_H: \hat{p}_H(t + k), \quad (2.22)$$

де T_H – горизонт прогнозування у часі (time horizon), що визначає кількість дискретних кроків у майбутньому, для яких здійснюється передбачення положення людини;

$\hat{p}_H(t + k)$ – прогнозоване положення (позиція) людини у момент часу $t + k$, де $k \in [1, T_H]$;

t – поточний момент часу, від якого починається прогнозування;

k – індекс кроку прогнозу в межах горизонту T_H , використовується для ітераційного передбачення траєкторії;

\hat{p}_H – символ «^» вказує на те, що це не реальна позиція, а оцінка або прогноз, отриманий з моделі руху (наприклад, моделі CV – constant velocity або з використанням фільтра Калмана);

– зона безпеки (комфорту) людини:

$$R_{safe}(t) = R_0 + \alpha \|\dot{p}_H(t)\|, \quad (2.23)$$

де $R_{safe}(t)$ – радіус зони безпеки (комфорту) людини у момент часу t , що визначає мінімальну відстань, на якій робот може наблизитися до людини без порушення комфортної взаємодії;

R_0 – базовий (статичний) радіус зони безпеки, що враховує особистий простір людини, навіть коли вона нерухома;

α – коефіцієнт пропорційності, що визначає, наскільки швидкість людини впливає на розмір зони безпеки; більші значення α збільшують зону при високих швидкостях руху;

$\|\dot{p}_H(t)\|$ – норма вектора швидкості людини в момент часу t , що відображає інтенсивність її руху; чим швидше рухається людина, тим більша зона безпеки.

Модель (2.23) дозволяє адаптивно змінювати розмір зони комфорту залежно від поведінки людини;

– поле ризику, це гаусівське поле ризику навколо прогнозованого положення людини, де ризик спадає з віддаленням за Махаланобісовою відстанню:

$$\mathcal{R}(x, y) = \exp\left(-\frac{1}{2}(x - \hat{p}_H(t))^T \Sigma_T^{-1}(t)(x - \hat{p}_H(t))\right), \quad (2.24)$$

де $\mathcal{R}(x, y)$ – безрозмірна інтенсивність ризику в планарній точці $x = [x, y]^T$, тобто чим ближче до людини, тим вища величина (максимум 1 у центрі);

$x = [x, y]^T$ – поточна 2D-позиція у робочій площині, в якій оцінюємо ризик; використовується в плануванні траєкторій EE/ланок;

$\hat{p}_H(t) \in \mathbb{R}^2$ – прогнозована позиція людини в момент часу t , отримана з моделі прогнозу/фільтра;

$(x - \hat{p}_H(t))^T \Sigma_T^{-1}(t)(x - \hat{p}_H(t))$ – квадрат Махаланобісової відстані, що вимірює «скільки сигм» точка x від центру з урахуванням напрямної невизначеності;

$\exp(\cdot)$ – множина, що задає гладке спадання ризику; відсутність нормувального коефіцієнта означає, що це відносне поле ризику/вартість, придатне як додаткова частина цільової функції в QP/MPC або як ваговий шар у CBF-обмеженнях.

Локальна задача керування Quadratic Programming/ Model Predictive Control (QP/MPC) на швидкостях, є така, що на кожному кроці для робота i розв'язується QP:

$$\min_{\dot{q}_i} \|J_{p,j}\dot{q}_i - v_i^{task}\|_{Q_v}^2 + \|\dot{q}_i\|_{R_\tau}^2 + \lambda_\mu(\mu_* - \mu_i(q_i))_+^2 + \lambda_\xi \|J_{p,i}\dot{q}_i - \hat{v}_{N_i}\|^2, \quad (2.21)$$

де \dot{q}_i – вектор швидкостей зчленувань робота i , змінна оптимізації, яку потрібно знайти;

$J_{p,j}$ – позиційна частина матриці Якобіана для робота i , що відображає зв'язок між швидкістю зчленувань і лінійною швидкістю енд-ефектора;

v_i^{task} – цільова швидкість робочого органа (енд-ефектора) для виконання конкретного завдання (наприклад, слідування траєкторії);

$\|J_{p,j}\dot{q}_i - v_i^{task}\|_{Q_v}^2$ – перший член функціоналу, який мінімізує відхилення від бажаної швидкості енд-ефектора з ваговою матрицею Q_v , що визначає важливість цього завдання;

$\|\dot{q}_i\|_{R_\tau}^2$ – другий член, який мінімізує енергію або момент керування (через швидкості), із ваговою матрицею R_τ , що забезпечує регуляризацію для згладження руху;

$\lambda_\mu (\mu_* - \mu_i(q_i))_+^2$ – штраф за порушення обмеження маніпулябельності;

$\mu_i(q_i)$ – показник маніпулябельності робота i ;

μ_* – мінімально допустиме значення маніпулябельності;

$(\cdot)_+$ – означає позитивну частину (штраф активується, якщо $\mu_i < \mu_*$);

λ_μ – коефіцієнт ваги для цього штрафу;

$\lambda_\xi \|J_{p,i} \dot{q}_i - \hat{v}_{N_i}\|^2$ – терм для узгодження швидкості з середнім значенням швидкостей сусідів у комунікаційному графі;

\hat{v}_{N_i} – прогнозована або усереднена швидкість сусідніх роботів N_i ;

λ_ξ – вага цього узгодження, що визначає важливість координації.

2.4 Розробка метода децентралізованого керування групою колаборативних роботів-маніпуляторів

Розробка методу децентралізованого керування групою колаборативних роботів-маніпуляторів є необхідною через зростаючу складність виробничих процесів у межах концепції Індустрії 5.0, де передбачається тісна взаємодія роботів між собою та з людьми. Централізовані алгоритми в таких системах мають суттєві обмеження, зокрема високу затримку обміну даними, вразливість до відмови центрального вузла та погану масштабованість. Децентралізований підхід забезпечує автономність кожного робота, дозволяє швидко адаптуватися до зміни конфігурації робочого середовища, уникати зіткнень та узгоджувати дії без єдиного контролера. Такий метод базується на локальних обчисленнях з обмеженим обміном інформацією, що знижує навантаження на мережу і підвищує стійкість системи до збоїв. Пропонований алгоритм передбачає розподілене формування цільових траєкторій, використання бар'єрних функцій безпеки та локальної оптимізації через QR або MPC, що гарантує дотримання обмежень у реальному часі. Його реалізація включає покроковий процес: побудова локальної моделі середовища, прогноз динамічних перешкод, визначення зони безпеки, розрахунок оптимальної швидкості або переміщення та координація з сусідніми агентами через мінімальний обмін даними. Такий

підхід дозволяє досягти баланс між безпекою, ефективністю та когнітивною взаємодією в умовах невизначеності та високої динаміки виробничих процесів. Представимо метод покрокова, це обловлено тим, що на базі даного метода потрібно буде розробити алгоритм та програму для численного моделювання.

Вхід (для кожного робота i): q_i, \dot{q}_i ; локальні сенсори; карти \mathcal{M}_i ; сусіди N_i ; людська поза/швидкість u_H з оцінкою (\hat{y}_H, P_H) .

Крок 0 (ініціалізація):

- калібрувати M_i, C_i, g_i (або ідентифікувати з PEM);
- налаштувати $M_{d,i}, D_{d,i}, K_{d,i}, Q_v, R_\tau, \lambda_\xi, \gamma, d_m, d_{min}, R_0, \alpha$;
- встановити $R_c, \eta, H, \Delta t$.

Крок 1 (перцепція й ф'южн):

– оновити білефи $\hat{p}_H, \dot{p}_H, P_H$, карту $\phi(\cdot, t)$ з невизначеністю та локальний SDF-фрагмент ϕ_i .

Крок 2 (планування мікрозадачі):

– згенерувати x_i^{ref} / v_i^{task} з урахуванням локального завдання (підхопити деталь, утримати позу тощо) + м'які узгодження з trajectory-sketch сусідів.

Крок 3 (побудова CBF-обмежень):

- сформулювати $h_{ij}, h_{i,H}, h_{iO}$ і відповідні лінійні нерівності по \dot{q}_i .

Крок 4 (децентралізована оптимізація):

- розв'язати локальне QR (швидкісний або моментний рівень);
- 1–2 ітерації обміну ξ_i (консенсус) у межах циклу реального часу.

Крок 5 (виконання й імпеданс):

– задати $q_i^* \rightarrow$ інтегрувати q_i , або обчислити τ_i через імпеданс/інверсну динаміку;

– моніторити контакти: якщо $F_i^{ext} >$ порогу – зменшити $K_{d,i}$ (підвищити «м'якість»).

Гарантії безпеки: при коректно обраній γ і побудові $h(\cdot)$ множини $\{h \geq\}$ є інваріантними \Rightarrow дотримання відстаней до людини, роботів і перешкод у реальному часі.

Запропонований метод децентралізованого керування для групи колаборативних роботів-маніпуляторів має суттєві переваги, що роблять його ефективним для реалізації чисельного моделювання у межах моделі мультиагентних систем. Головною перевагою є відсутність єдиного центру управління, що забезпечує високу надійність і стійкість до відмов, адже кожен робот приймає рішення локально, враховуючи інформацію від сусідів і прогнозовані дії людини. Такий підхід дозволяє знизити затримки в обміні даними, що особливо критично в динамічних умовах, де необхідна швидка реакція на зміни середовища. Метод забезпечує масштабованість системи, оскільки додавання нових роботів не вимагає перебудови загальної архітектури керування. Використання комунікаційного графа гарантує оптимальну взаємодію між агентами без надмірного обміну інформацією, а локальні задачі оптимізації на основі QP/MPC дозволяють враховувати бар'єрні функції безпеки та зони комфорту людини в реальному часі. Крім того, алгоритм адаптується до непередбачуваних змін, таких як поява нових об'єктів або переміщення оператора, завдяки прогнозу траєкторій та гнучкому коригуванню локальних рішень. У результаті метод поєднує безпеку, ефективність і гнучкість, що робить його придатним для чисельного моделювання складних сценаріїв у виробничих умовах Індустрії 5.0.

2.5 Висновки до 2 розділу

У другому розділі було розроблено математичне забезпечення, яке забезпечує можливість моделювання децентралізованого керування групою колаборативних роботів-маніпуляторів у спільній робочій зоні з людиною. Створена модель робота-маніпулятора враховує кінематичні та динамічні параметри, що дозволяє формувати точні описи руху для виконання складних маніпуляційних задач. Модель представлення навколишнього середовища включає об'єкти, динамічні перешкоди та зону безпеки оператора, що є критично важливим для дотримання вимог Індустрії 5.0 та безпечної взаємодії людини і

робота. У рамках моделі децентралізованого керування було реалізовано мультиагентний підхід із застосуванням локальних задач оптимізації, що гарантує масштабованість і стійкість системи до відмов.

Розроблений метод децентралізованого керування передбачає інтеграцію бар'єрних функцій безпеки, прогнозу траєкторій людини та обмежень комфортної зони, що дозволяє ефективно уникати зіткнень і підтримувати стабільну роботу колективу роботів. Отримане математичне забезпечення створює основу для побудови алгоритмів керування, орієнтованих на адаптивність, надійність і ефективність у складних динамічних виробничих сценаріях.

3 РОЗРОБКА ПРОГРАМИ ДЛЯ МОДЕЛЮВАННЯ ДЕЦЕНТРАЛІЗОВАНОГО КЕРУВАННЯ ГРУПОЮ КОЛАБОРАТИВНИХ РОБОТІВ-МАНІПУЛЯТОРІВ

3.1 Вибір мови програмування та середовища розробки

Вибір мови програмування та середовища розробки є ключовим етапом при реалізації програми численного моделювання децентралізованого керування групою колаборативних роботів-маніпуляторів у єдиній робочій зоні з людиною. Серед існуючих високорівневих мов програмування, таких як C++, Java, MATLAB та Python, кожна має свої переваги та обмеження. C++ забезпечує високу швидкодію та можливість роботи з низькорівневими структурами, однак має складний синтаксис і значні витрати часу на розробку. Java вирізняється переносимістю та багатопотоковістю, але поступається у швидкості виконання та потребує більше ресурсів. MATLAB традиційно використовується для наукових розрахунків і моделювання, однак є закритою комерційною платформою, що обмежує масштабованість і вимагає високих ліцензійних витрат.

Python, на відміну від зазначених мов, поєднує простоту синтаксису, потужну екосистему бібліотек та високу гнучкість, що робить його оптимальним вибором для науково-технічних досліджень і чисельного моделювання. Для задач моделювання мультиагентних систем і керування роботами Python надає широкий набір інструментів: NumPy, SciPy для лінійної алгебри та оптимізації; Matplotlib, Plotly для візуалізації результатів; SymPy для аналітичних обчислень; а також бібліотеки штучного інтелекту (TensorFlow, PyTorch) для реалізації адаптивних алгоритмів. Додатковою перевагою Python є інтеграція з ROS (Robot Operating System) через пакети rospy, що забезпечує можливість створення програмного забезпечення для робототехнічних систем у реальному часі.

Щодо вибору середовища розробки, PyCharm 2025.1.1.1 є найбільш доцільним рішенням завдяки своїй функціональності та орієнтації на роботу з великими науковими проєктами. PyCharm забезпечує інтелектуальну підтримку коду (code completion, рефакторинг), потужні інструменти налагодження та тестування, а також інтеграцію з системами контролю версій (Git), що є критично важливим для командної роботи над складними системами. Крім того, нова версія PyCharm підтримує розширені можливості роботи з віртуальними середовищами, що дозволяє гнучко керувати залежностями проєкту, а також інтеграцію з хмарними сервісами та Docker для контейнеризації обчислювальних процесів. У поєднанні з підтримкою Jupyter Notebook, PyCharm дає можливість створювати інтерактивні середовища для тестування моделей та візуалізації даних.

В таблиці 3.1 приведено порівняльний аналіз мов високого рівня програмування для розробки програми численного моделювання децентралізованого керування групою колаборативних роботів-маніпуляторів у єдиній робочій зоні з людиною на основі моделі мультиагентних систем

Таблиця 3.1 – Порівняльний аналіз мов високого рівня програмування

Критерій	Python	C++	Java
1	2	3	4
Простота синтаксису	Дуже проста, інтуїтивна, підходить для швидкої розробки	Складний синтаксис, вимагає більше часу на реалізацію	Простий, але більш громіздкий у порівнянні з Python
Швидкодія	Нижча, ніж у C++, але оптимізується бібліотеками (NumPy, Numba)	Висока продуктивність, особливо для обчислювально складних задач	Середня, залежить від JVM
Бібліотеки для робототехніки	Дуже велика кількість (ROS, NumPy, SciPy, TensorFlow, PyTorch)	Менша кількість, складніша інтеграція	Обмежені можливості у порівнянні з Python

Продовження таблиці 3.1

1	2	3	4
Зручність для мультиагентних систем	Висока: підтримка симуляційних пакетів (Mesa, SimPy), інтеграція з AI	Складно реалізувати спеціальних фреймворків без	Потребує додаткових бібліотек, складніше ніж у Python
Інтеграція з ROS	Легка (rospy, rosbridge)	Є підтримка (roscpp), але складніше налаштувати	Обмежена підтримка

Python виграє за простотою розробки, наявністю бібліотек для чисельного моделювання, мультиагентних систем, AI та інтеграції з ROS. C++ підходить для високопродуктивних реальних систем, але вимагає більше часу та складної оптимізації. Java добре підходить для корпоративних рішень, але менш ефективна для швидкої розробки прототипів і наукового моделювання.

Таким чином, поєднання Python і PyCharm 2025.1.1.1 дозволяє реалізувати ефективну, гнучку та масштабовану програмну платформу для чисельного моделювання децентралізованого керування мультиагентними системами роботів. Такий вибір забезпечує скорочення часу розробки, зручність тестування алгоритмів, адаптивність до нових методів та відповідність сучасним вимогам Індустрії 5.0.

3.2. Розробка алгоритму програми для моделювання децентралізованого керування групою колаборативних роботів-маніпуляторів

Розробка алгоритму роботи програми для чисельного моделювання децентралізованого керування групою колаборативних роботів-маніпуляторів у єдиній робочій зоні є необхідною через складність координації багатьох агентів у реальному часі. У виробничих умовах роботи повинні взаємодіяти між собою та з людиною, уникаючи зіткнень і забезпечуючи безпечні дистанції, що потребує динамічної адаптації траєкторій. Без алгоритмічної основи неможливо врахувати змінні фактори середовища, такі як переміщення об'єктів, поява нових

перешкод або зміни позицій інших роботів. Чисельне моделювання дозволяє перевіряти різні сценарії та методи координації без ризику для обладнання та персоналу, що значно знижує витрати та підвищує безпеку. Алгоритм також забезпечує можливість впровадження мультиагентних стратегій на основі децентралізованих систем, які є більш стійкими до відмов порівняно з централізованими підходами. Розроблений алгоритм (рис. 3.1) є основою для побудови програмного забезпечення, здатного виконувати оптимізацію маршруту, прогнозування руху та оцінку ризиків, що критично для інтеграції колаборативних роботів у складні виробничі процеси.

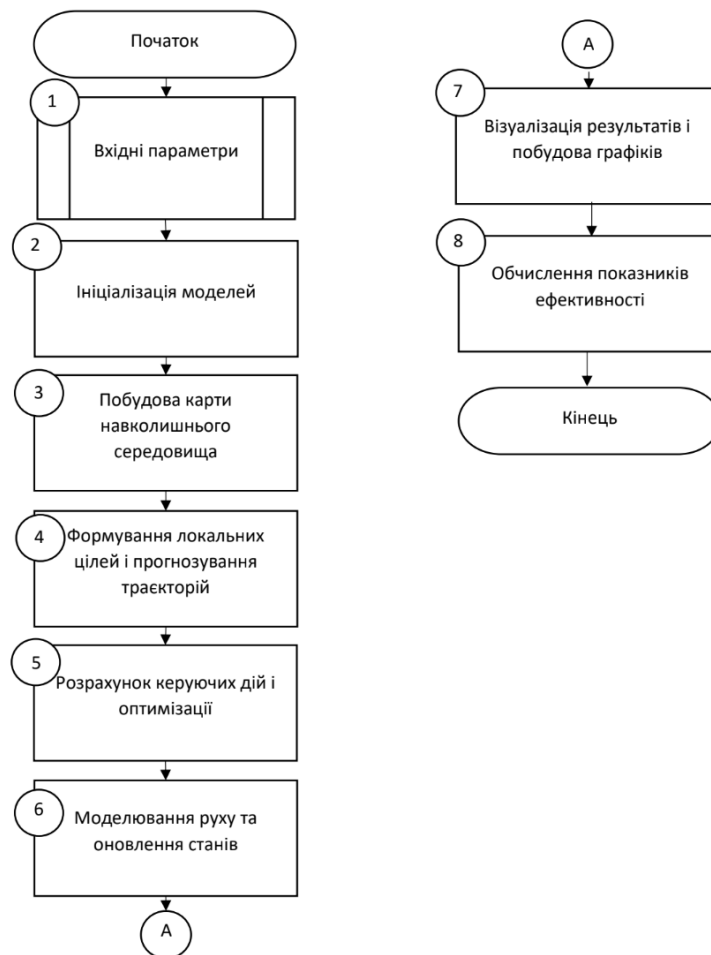


Рисунок 3.1 – Алгоритм програми для моделювання децентралізованого керування групою колаборативних роботів-маніпуляторів

Крок 1. Вхідні параметри. На початковому етапі алгоритм отримує набір вхідних параметрів, що визначають сценарій моделювання. До них належать

кількість роботів, початкові та цільові координати кожного робота, розташування статичних перешкод, межі робочої зони, максимальна швидкість руху (4 м/с) та часовий горизонт симуляції. Ці дані дозволяють сформувати середовище для побудови траєкторій у рамках мультиагентної системи.

Крок 2. Ініціалізація моделей. На другому етапі здійснюється ініціалізація математичних моделей, що описують колаборативних роботів-маніпуляторів та децентралізоване керування. Кожен агент отримує власну модель локального планування з урахуванням обмежень швидкості, зони безпеки відносно людини та передбачення руху інших агентів. Ініціалізація також включає формування поля ризику і зон комфорту.

Крок 3. Побудова карти навколишнього середовища. Алгоритм створює карту робочої зони, де фіксуються перешкоди та визначаються безпечні області для руху роботів. Середовище розглядається як дискретизоване поле з можливістю розрахунку вартості переміщення для кожного вузла з урахуванням ризику зіткнення та зон комфорту людини.

Крок 4. Формування локальних цілей і прогнозування траєкторій. На основі заданих глобальних цілей для кожного робота визначаються проміжні точки руху та прогнозується траєкторія на основі моделі децентралізованого керування. Для цього враховується передбачуваний рух інших агентів та обхід перешкод, що забезпечує безконфліктне переміщення.

Крок 5. Розрахунок керуючих дій і оптимізації. На цьому етапі для кожного робота вирішується локальна задача оптимізації швидкостей і напрямків руху з урахуванням обмежень по швидкості, зон безпеки та мінімізації ризику зіткнення. Застосовуються методи мультиагентної координації для узгодження траєкторій.

Крок 6. Моделювання руху та оновлення станів. Роботи здійснюють переміщення відповідно до розрахованих керуючих дій. На кожному кроці оновлюється їх положення, прогнозуються нові траєкторії, а також відбувається облік зміни ризикових зон. У випадку наближення до перешкоди або іншого агента відбувається корекція траєкторії.

Крок 7. Візуалізація результатів і побудова графіків. По завершенні симуляції або на проміжних етапах алгоритм будує графіки: траєкторії руху кожного робота, карти розташування перешкод, зміни швидкості з часом, динаміку відстані між агентами та час досягнення цілей. Візуалізація виконується за допомогою бібліотеки Matplotlib.

Крок 8. Обчислення показників ефективності. Після завершення моделювання обчислюється загальний час досягнення цілей, середня швидкість руху, кількість конфліктних ситуацій, відстань до людини у критичних моментах та оцінка безпечності руху. Це дозволяє зробити висновки про ефективність розробленого методу.

Розроблений алгоритм програми для моделювання децентралізованого керування групою колаборативних роботів-маніпуляторів має низку суттєвих переваг, що забезпечують його ефективність у складних виробничих сценаріях. Основною перевагою є здатність системи до масштабованості, що дозволяє легко додавати нових роботів без суттєвого зростання обчислювальної складності. Алгоритм підтримує високу гнучкість та адаптивність до змін у навколишньому середовищі, зокрема до появи динамічних перешкод і зміни положення людини, що гарантує безпеку роботи. Використання децентралізованого підходу мінімізує залежність від єдиного центра керування, підвищуючи надійність та стійкість системи до відмов окремих агентів. Крім того, алгоритм забезпечує можливість оптимізації траєкторій у реальному часі з урахуванням обмежень швидкості, зони комфорту людини та уникнення зіткнень. Його архітектура сумісна з мультиагентними моделями та легко інтегрується з алгоритмами прогнозування на основі штучного інтелекту, що розширює функціонал для складних сценаріїв. Моделювання дозволяє зменшити витрати на фізичні експерименти, знижує ризики аварійних ситуацій та підвищує ефективність підготовки виробничих процесів. Таким чином, алгоритм є важливим елементом реалізації концепції Industry 5.0 та розвитку когнітивних робототехнічних систем.

3.3 Розробка функцій CBF

CBF (Control Barrier Function) у контексті моделювання децентралізованого керування групою колаборативних роботів-маніпуляторів визначається як математичний інструмент, що використовується для забезпечення безпеки системи під час руху в загальній робочій зоні. Основна ідея полягає у формуванні функцій бар'єрів, які обмежують стани системи таким чином, щоб уникати зіткнень між роботами, перешкодами та людиною, зберігаючи при цьому досяжність цільових позицій. У децентралізованому підході CBF дозволяє кожному агенту самостійно розраховувати допустимі траєкторії на основі локальної інформації та взаємодії з іншими агентами без потреби в централізованому контролері. Це забезпечує гарантовану безпечну поведінку навіть у динамічному середовищі з непередбачуваними змінами. Таким чином, CBF є ключовим елементом для інтеграції адаптивних і безпечних стратегій керування у мультиагентні системи.

Приведемо приклад програмної реалізації CBF:

1. CBF між роботами:

```
for j in range(NUM_ROBOTS):
    if j == i: continue
    pj = positions[j]
    rij = ROBOT_RADIUS + ROBOT_RADIUS +
INTER_ROBOT_BUFFER
    dp = p - pj
    h_ij = dp @ dp - (rij) ** 2
    # ∇_p h = 2 (p - p_j)
    grad_h = 2.0 * dp
    # constraint: grad_h^T v >= -gamma * h
    a = grad_h
    b = -CBF_GAMMA * h_ij
    A_list.append(a)
```

```
b_list.append(b)
```

Цей фрагмент коду реалізує обчислення обмежень безпеки на основі функцій бар'єрів CBF для уникнення зіткнень між роботами у мультиагентній системі. У циклі перебираються всі роботи, крім поточного, і для кожної пари визначається відстань та вектор різниці позицій. Обчислюється значення бар'єрної функції h_{ij} , яке гарантує мінімальну відстань між роботами, та її градієнт, необхідний для формування лінійних обмежень. Далі створюються коефіцієнти a і b для нерівності, що забезпечує умову $\nabla h^T \geq -\gamma h$, яка гарантує, що рух робота не зменшить відстань до критичного рівня. Отримані значення додаються до списків обмежень, які пізніше використовуються в оптимізаційній задачі для корекції швидкості роботів.

2. CBF для людини:

```
dpiH = p - human_pos
```

```
h_iH = dpiH.T @ np.linalg.inv(human_Sigma) @ dpiH - R_s ** 2
```

```
grad_hH = 2.0 * (np.linalg.inv(human_Sigma) @ dpiH) # gradient wrt p
```

```
aH = grad_hH
```

```
bH = -CBF_GAMMA * h_iH
```

```
A_list.append(aH)
```

```
b_list.append(bH)
```

Цей фрагмент коду реалізує обчислення бар'єрної функції для забезпечення безпечної відстані між роботом і людиною в робочій зоні. Спочатку визначається вектор різниці між позицією робота та людини, після чого обчислюється значення бар'єрної функції h_{iH} , яка враховує просторову невизначеність положення людини через матрицю коваріації. Потім розраховується градієнт цієї функції відносно позиції робота, що необхідно для формування обмежень у вигляді лінійної нерівності. На основі цього створюються коефіцієнти a_H та b_H , які описують умову безпечного руху робота з урахуванням допустимої зони навколо людини. Ці значення додаються до загального списку обмежень для подальшої оптимізації управління.

3. SDF перешкоди:

```

phi = sdf_at_point(p) # відстань до найближчої перешкоди (позитивна поза)
grad_phi = grad_sdf(p)
# h_iO = phi - d_min >= 0 -> grad_phi^T v >= -gamma * h_iO
d_min = ROBOT_RADIUS + 0.2
h_iO = phi - d_min
aO = grad_phi
bO = -CBF_GAMMA * h_iO
A_list.append(aO)
b_list.append(bO)

```

Цей фрагмент коду використовується для забезпечення уникнення зіткнення робота з перешкодами за допомогою бар'єрних функцій. Спочатку обчислюється відстань від поточного положення робота до найближчої перешкоди за допомогою функції підписаної відстані, а також її градієнт, що визначає напрямок до безпечної області. На основі цього розраховується значення бар'єрної функції h_{iO} , яке показує, наскільки робот дотримується мінімально допустимої відстані до перешкоди. Потім формується лінійне обмеження для швидкості робота у вигляді коефіцієнтів a_o та b_o , які гарантують, що рух буде безпечним і не приведе до зіткнення. Ці параметри додаються до списку обмежень, що враховуються під час оптимізації управління.

3.4 Розробка функцій численного моделювання

При розробці програмного коду для побудови карт місцевості з метою моделювання децентралізованого керування групою колаборативних роботів-маніпуляторів у спільній робочій зоні з людиною необхідно врахувати кілька критичних аспектів. По-перше, карта повинна відображати динамічність середовища, оскільки положення робітників, об'єктів та самих роботів може змінюватися в реальному часі, що вимагає адаптивного оновлення карти. По-друге, важливо забезпечити точну інтеграцію даних від кількох сенсорів, таких як камери, ультразвукові датчики та ІМУ, для підвищення точності просторового

представлення. Також потрібно реалізувати багаторівневу структуру карти, яка включає статичні елементи середовища, зони безпеки для людини та прогнозовані траєкторії руху роботів. Особливу увагу слід приділити представленню зон комфорту людини та створенню буферних областей, що запобігають порушенню безпечної дистанції. Крім того, система повинна бути оптимізована для роботи в умовах обмежених обчислювальних ресурсів, враховуючи одночасну взаємодію декількох агентів. Не менш важливо забезпечити можливість інтеграції алгоритмів CBF та MPC для обробки обмежень у просторі карти. Усі ці особливості повинні бути враховані для досягнення високого рівня безпеки та ефективності у децентралізованому керуванні.

```
# Ініціалізація станів роботів
positions = starts.copy().astype(float)    # shape (N,2)
velocities = np.zeros_like(positions)     # поточні швидкості
reached = np.zeros(NUM_ROBOTS, dtype=bool)
arrival_time = np.full(NUM_ROBOTS, np.nan)

# Для логування
traj_logs = [ [positions[i].copy()] for i in range(NUM_ROBOTS) ]
speed_logs = [ [] for _ in range(NUM_ROBOTS) ]
dist_obst_logs = [ [] for _ in range(NUM_ROBOTS) ]
dist_human_logs = [ [] for _ in range(NUM_ROBOTS) ]
consensus_error_logs = [ [] for _ in range(NUM_ROBOTS) ]

# Вспомогательные: найти соседей по радиусу (геометрично)
def neighbors_of(i, positions, R_comm):
    ni = []
    for j in range(len(positions)):
        if j == i: continue
        if norm(positions[j] - positions[i]) <= R_comm:
            ni.append(j)
    return ni
```

```

# Симуляція крок за кроком
time_axis = np.arange(0, T_TOTAL, DT)
for step, t in enumerate(time_axis):
    # оновити модель людини (проста лінійна модель)
    human_pos = human_pos + human_vel * DT
    # clamp в межі карти
    human_pos = np.clip(human_pos, 0.0, GRID_SIZE - 1.0)
    human_speed = norm(human_vel)
    R_s = R_safe(human_speed)

```

Цей фрагмент коду реалізує початкову ініціалізацію станів колаборативних роботів-маніпуляторів і підготовку до симуляції децентралізованого керування у спільній робочій зоні. Масиви `positions` і `velocities` зберігають поточні координати та швидкості роботів, тоді як масиви `reached` і `arrival_time` відстежують факт досягнення цільової точки та час прибуття кожного робота. Для аналізу динаміки руху створюються журнали (`traj\logs`, `speed\logs`, `dist\obst\logs`, `dist\human\logs`, `consensus\error\logs`) для запису траєкторій, швидкостей, відстаней до перешкод, людини та помилок консенсусу між агентами. Функція `neighbors_of` визначає сусідів робота у межах радіусу зв'язку, що критично для реалізації децентралізованого алгоритму взаємодії. Основний цикл симуляції оновлює положення людини за лінійною моделлю руху, обмежує її координати в межах карти та обчислює безпечний радіус взаємодії залежно від швидкості людини. Цей блок є базовим для подальшої реалізації алгоритмів керування та забезпечення безпеки в динамічному середовищі.

```

# для кожного робота: формуємо бажану швидкість (до цілі) + узгодження з сусідами

```

```

desired_vs = np.zeros_like(velocities)
for i in range(NUM_ROBOTS):
    if reached[i]:
        desired_vs[i] = np.array([0.0, 0.0])

```

```

    continue
p = positions[i]
goal = goals[i]
to_goal = goal - p
dist_to_goal = norm(to_goal)
# якщо близько до цілі - зупинка та лог часу
if dist_to_goal < 0.8:
    reached[i] = True
    arrival_time[i] = t
    desired_vs[i] = np.zeros(2)
    continue
# бажана швидкість до цілі (пропорційна, але обмежена)
v_goal = unit(to_goal) * min(MAX_SPEED, dist_to_goal / 0.5) #
пропорція для плавності
# консенсус: усереднена швидкість сусідів (як ескіз траєкторії)
neigh = neighbors_of(i, positions, COMM_RADIUS)
if len(neigh) > 0:
    v_neighbors = np.mean(velocities[neigh], axis=0)
else:
    v_neighbors = np.zeros(2)
# комбінуємо (використовуємо LAMBDA_XI як вага для узгодження)
desired_vs[i] = (1 - LAMBDA_XI) * v_goal + LAMBDA_XI *
v_neighbors

```

Цей фрагмент коду реалізує формування бажаних швидкостей для кожного робота з урахуванням досягнення цілі та узгодження руху з сусідами, що є ключовим елементом децентралізованого керування. Якщо робот уже досяг цілі, його швидкість встановлюється нульовою, і фіксується час прибуття. Для активних роботів обчислюється напрямок до цілі та відстань, при цьому близькість до мети призводить до зупинки для уникнення коливань. Бажана швидкість визначається як пропорційна до відстані до цілі, але обмежується

максимально допустимим значенням для плавного руху. Для забезпечення колективної узгодженості враховується середня швидкість сусідів, визначених у радіусі зв'язку, що дозволяє уникати конфліктів траєкторій. Остаточна швидкість є зваженою комбінацією вектора до цілі та швидкості сусідів із використанням коефіцієнта $LAMBDA \setminus XI$. Це забезпечує баланс між досягненням власної цілі та колаборативною поведінкою у мультиагентному середовищі.

3.5 Висновки до 3 розділу

У третьому розділі було розроблено програмне забезпечення для моделювання децентралізованого керування групою колаборативних роботів-маніпуляторів, яке дозволяє досліджувати поведінку мультиагентної системи у єдиній робочій зоні з людиною. Для реалізації обрано мову програмування Python та середовище, що забезпечує роботу з бібліотеками чисельного аналізу, візуалізації та обробки даних. Розроблений алгоритм включає ініціалізацію станів роботів, визначення сусідів у межах радіуса зв'язку, формування бажаних швидкостей із використанням цільових точок і принципів консенсусу, а також моделювання руху з урахуванням обмежень та безпечних зон для людини. Окремо описано ключові функції, зокрема для обчислення консенсусних швидкостей, контролю досягнення цілей, логування траєкторій і швидкостей, що дозволяє аналізувати поведінку системи. У рамках чисельного моделювання реалізовано покрокову симуляцію, що включає оновлення позицій, обробку зіткнень, перевірку умов досягнення цілей та аналіз показників взаємодії. Запропоноване рішення демонструє ефективність децентралізованого підходу, забезпечує адаптивність системи в динамічному середовищі та створює основу для подальшого впровадження у реальні виробничі сценарії.

4 ПРОВЕДЕННЯ ДОСЛІДЖЕНЬ ТА АНАЛІЗ ОТРИМАНИХ РЕЗУЛЬТАТІВ

4.1 Постановка задач експериментальних досліджень

Постановка задачі дослідження полягає у розробці, моделюванні та аналізі алгоритму децентралізованого керування групою колаборативних роботів-маніпуляторів у єдиній робочій зоні з людиною на основі принципів мультиагентних систем.

Основною проблемою є забезпечення узгодженої поведінки роботів при виконанні індивідуальних та колективних завдань, уникнення зіткнень між агентами та людиною, а також досягнення цільових точок із дотриманням обмежень по швидкості та зоні безпеки.

Мета дослідження полягає у створенні програмної моделі, яка відтворює реальні умови взаємодії роботів і дозволяє оцінити ефективність алгоритмів узгодження швидкостей, побудови траєкторій і забезпечення безпеки в обмеженому просторі.

У рамках моделювання було визначено кілька ключових задач дослідження:

- зміни координат роботів у часі, що дає можливість оцінити коректність побудови траєкторій і перевірити відсутність зіткнень між агентами та з людиною;
- швидкості руху кожного робота, що дозволяє оцінити дотримання обмежень по максимальній швидкості та ефективність адаптації алгоритму в різних сценаріях;
- динаміку відстаней між роботами і людиною, що дозволяє визначити рівень безпеки та правильність роботи механізму уникнення небезпечного зближення;

– часу досягнення цілей усіма агентами, що дозволяє оцінити загальну ефективність алгоритму узгодженого руху та оптимальність шляху.

Очікувані результати дослідження включають підтвердження стабільності децентралізованого керування, відсутність зіткнень, дотримання швидкісних обмежень та досягнення високого рівня синхронізації між агентами.

Для проведення чисельного моделювання використовується персональний комп'ютер із процесором не нижче Intel Core i7, оперативною пам'яттю 16 ГБ та графічним прискорювачем для обробки візуалізації. Як програмне забезпечення застосовується мова Python з бібліотеками NumPy, Matplotlib та SciPy для реалізації алгоритмів та побудови графіків, а також середовище Jupyter Notebook для інтерактивного аналізу результатів. Такий підхід дозволяє отримати повну картину поведінки системи, оцінити робастність запропонованого алгоритму та сформулювати висновки щодо його ефективності та можливості впровадження у реальні виробничі сценарії.

4.2 Проведення моделювання децентралізованого керування групою колаборативних роботів-маніпуляторів

Розмір дискретної карти задано як $GRID\backslash SIZE = 50$, що означає сітку 50×50 клітин для побудови SDF та візуалізації, забезпечуючи баланс між точністю і швидкістю обчислень.

Часовий крок інтегрування $DT = 0,1$ с визначає дискретизацію симуляції і дозволяє стабільно інтегрувати рухи роботів при реальному часовому масштабі.

Загальний час моделювання встановлено $T\backslash TOTAL = 40,0$ с, що дає можливість спостерігати поведінку системи на горизонті у 40 секунд; це відповідає $STEPS = \text{int}(T\backslash TOTAL/DT) = 400$ ітераціям симулятора.

У моделі використовується $NUM\backslash ROBOTS = 4$ роботів, що дозволяє вивчати взаємодію невеликої групи у мультиагентному сценарії.

Максимальна допустима швидкість кожного робота обмежена значенням $MAX\backslash SPEED = 4,0$ м/с для забезпечення фізично реалістичних рухів та безпеки.

Ефективний радіус робота встановлено як $ROBOT\backslash RADIUS = 0,8$ одиниць (умовних метрів карти), що використовується при побудові буферів навколо енд-ефектора.

Додатковий міжробочий буфер задано $INTER\backslash ROBOT\backslash BUFFER = 0,2$ одиниць для запасу безпеки при взаємних маневрах.

Параметр $CBF\backslash GAMMA = 2,0$ визначає «жорсткість» умови бар'єру – швидкість відштовхування від критичних меж у нерівності $\nabla h^T v \geq -\gamma h$

Кількість ітерацій послідовних проекцій для корекції швидкості обрана $PROJECT\backslash ITERS = 6$, що забезпечує збалансовану точність перетворення початкового вектора бажаної швидкості у безпечний.

Радіус комунікацій між агентами заданий $COMM\backslash RADIUS = 12,0$ одиниць карти, що визначає локальну мережу сусідів для консенсусу.

Вага узгодження швидкостей $LAMBDA\backslash XI = 0,8$ означає сильний вплив середньої швидкості сусідів при формуванні локальної бажаної швидкості.

У моделі задано три статичні перешкоди $OBSTACLES = [(18,26,5), (35,12,4), (30,35,6)]$, де кожна кортежна трійка – це центр (x, y) і радіус r у одиницях карти.

Початкові позиції роботів визначені як $starts = [[3.0,3.0], [3.0,46.0], [46.0,3.0], [20.0,40.0]]$, що розподіляє агентів по кутах і внутрішній точці карти.

Цільові позиції задані $goals = [[46.0,46.0], [46.0,3.0], [3.0,46.0], [40.0,6.0]]$, кожному роботу присвоєно індивідуальну мету для тестування деконфліктної координації.

Модель людини як динамічної перешкоди ініціалізована $human\backslash pos = [25.0,25.0]$, що розміщує її в центрі карти для максимального впливу на маршрути.

Швидкість людини $human\backslash vel = [0,05, 0,07]$ одиниць карти на секунду дає повільний рух оператора, еквівалентний $\sim 0,085$ одиниць/с у напрямку вектора швидкості.

Базовий радіус зони комфорту людини $R0 = 2,0$ одиниць задає мінімальний статичний особистий простір навколо людини.

Коефіцієнт чутливості R_{safe} задається $ALPHA = 1,5$, тому при зростанні швидкості людини зона комфорту розширюється пропорційно $\alpha \cdot \|v_H\|$.

Коваріаційна матриця невизначеності $human\backslash\Sigma = [[3.0, 0.5], [0.5, 2.0]]$ задає анізотропію ризик-поля людини: більший розкид по одній осі і кореляція 0,5.

Ці параметри разом задають умови задачі: дискретизацію простору і часу, динамічні обмеження агентів, геометрію перешкод і поведінку людини, на яких базуються всі функції алгоритму.

Параметри CBF і $PROJECT\backslash\text{ITERS}$ визначають, наскільки агресивно і точно система коригує бажані швидкості для забезпечення безпеки; при $CBF\backslash\text{GAMMA} = 2,0$ та 6 ітераціях досягається стабільна поведінка без надмірного ослаблення руху.

$COMM\backslash\text{RADIUS} = 12,0$ і $LAMBDA\backslash\text{XI} = 0,8$ забезпечують локальний консенсус, де робот враховує сильніше поведінку сусідів, що сприяє узгодженим, але не централізованим маневрам.

Параметри перешкод (позиції і радіуси) визначають складність навігації; задані центри і радіуси створюють перешкоди середнього розміру, що вимагають обхідних траєкторій.

Початкові й цільові координати формують тестову множину сценаріїв з перехресними маршрутами, що дозволяє перевірити алгоритм на запобігання блокуванню і колізіям.

Модель людини з $R_0 = 2,0$ та $ALPHA = 1,5$ разом із $human\backslash\Sigma$ визначає еліптичне ризик-поле, яке роботи враховують через Mahalanobis CBF, що імітує проксеміку реального оператора.

Очікувані результати моделювання включають коректні, безконфліктні траєкторії для всіх чотирьох роботів за горизонтом $T\backslash\text{TOTAL} = 40$ с, дотримання обмеження $MAX\backslash\text{SPEED} = 4,0$ м/с, відсутність порушень мінімальних дистанцій $ROBOT\backslash\text{RADIUS} + INTER\backslash\text{ROBOT}\backslash\text{BUFFER} = 1,8$ одиниць між агентами і підтримання відстані до людини не менше адаптивного R_{safe} . Отримані результати численного моделювання представлені на рисунках 4.1 – 4.6.

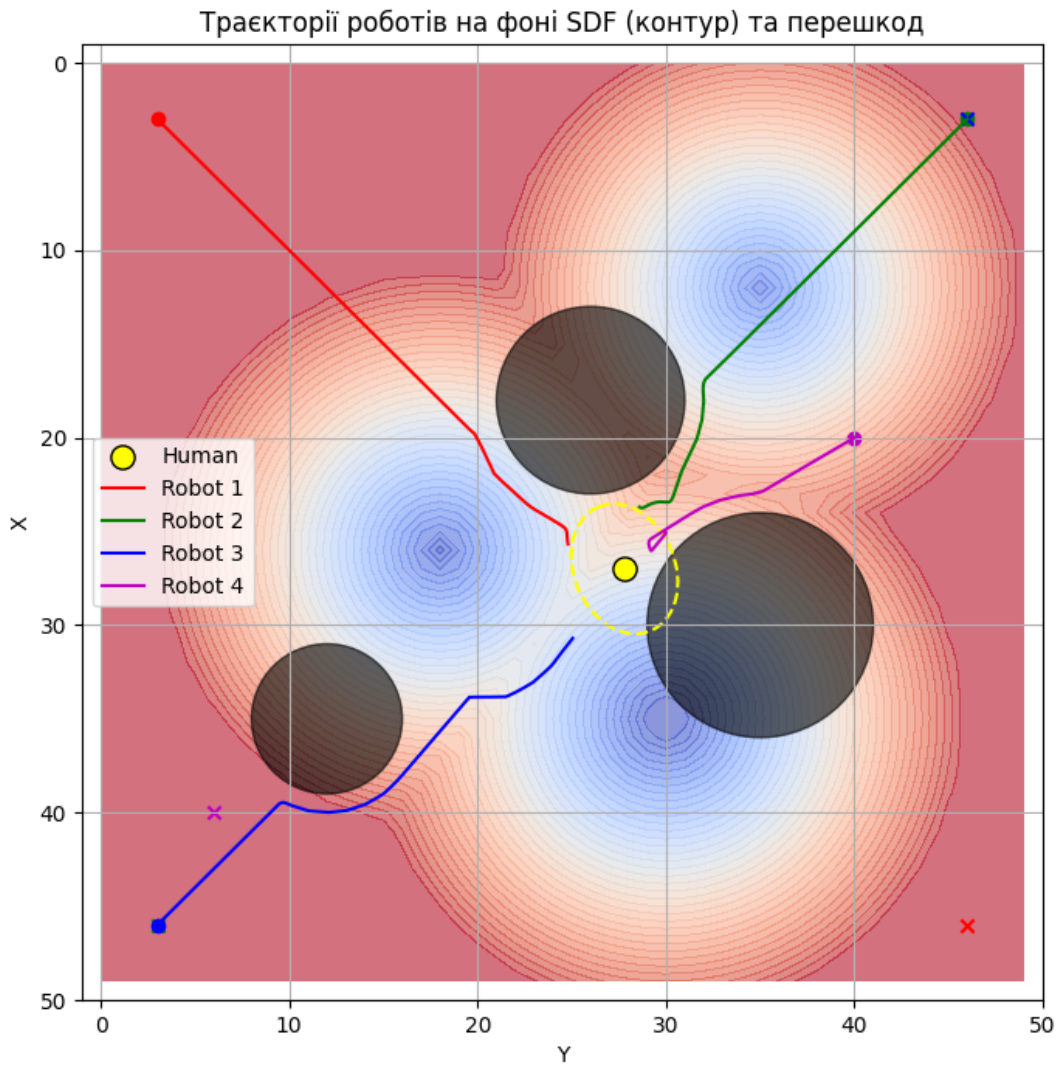


Рисунок 4.1 – Траєкторія роботів на фоні SDF та перешкод

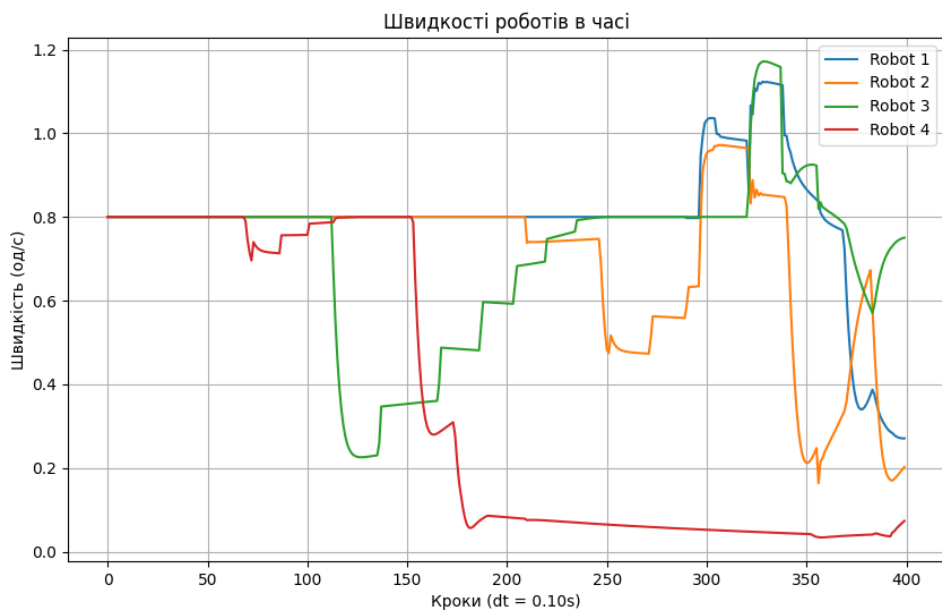


Рисунок 4.2 – Графік швидкості роботів в часі

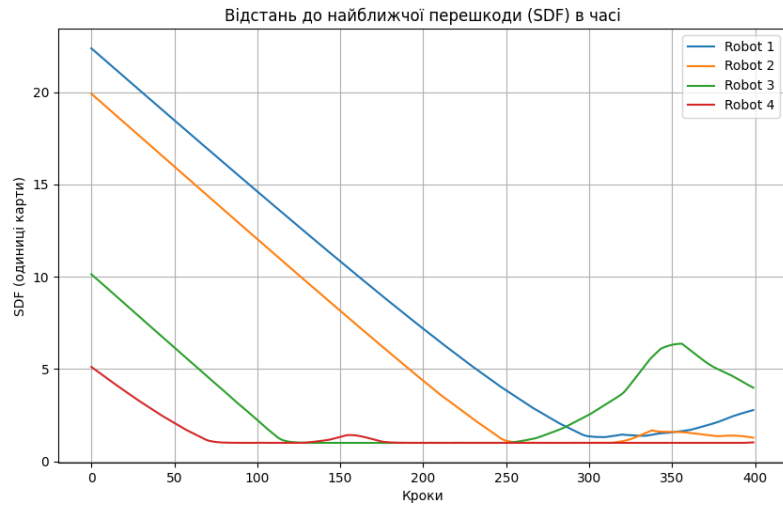


Рисунок 4.3 – Графік відстані до найближчої перешкоди в часі

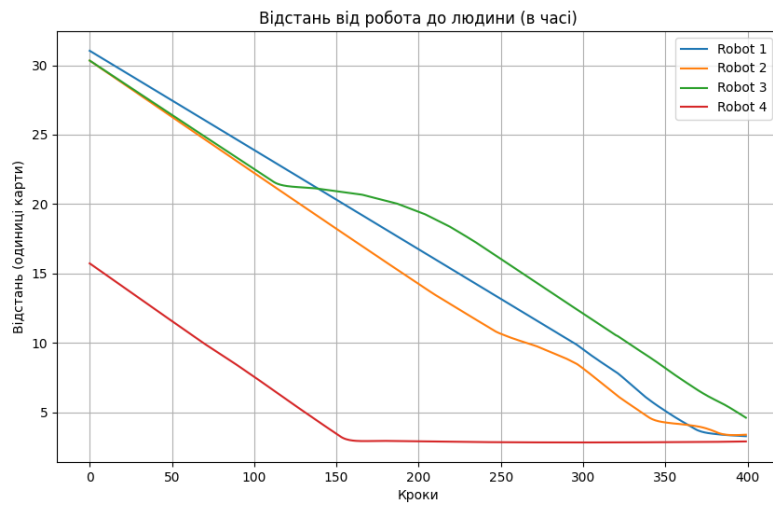


Рисунок 4.4 – Графік відстані від робота до людини в часі

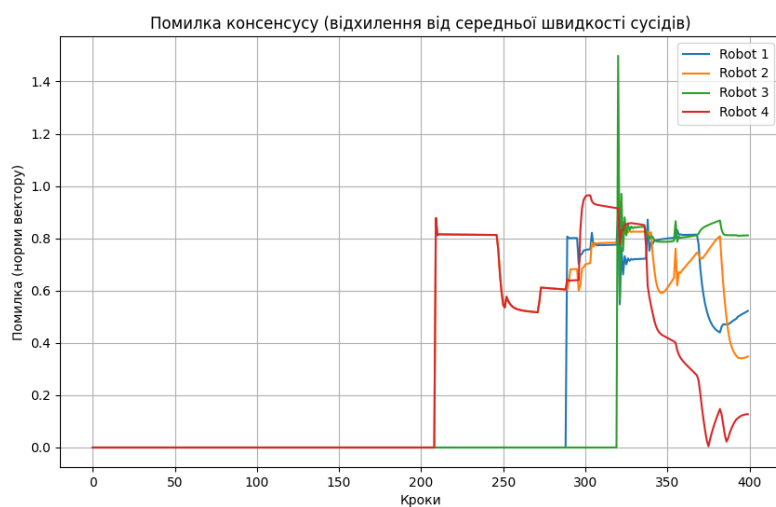


Рисунок 4.5 – Графік помилки консенсусу (відхилення від середньої швидкості сусідів)

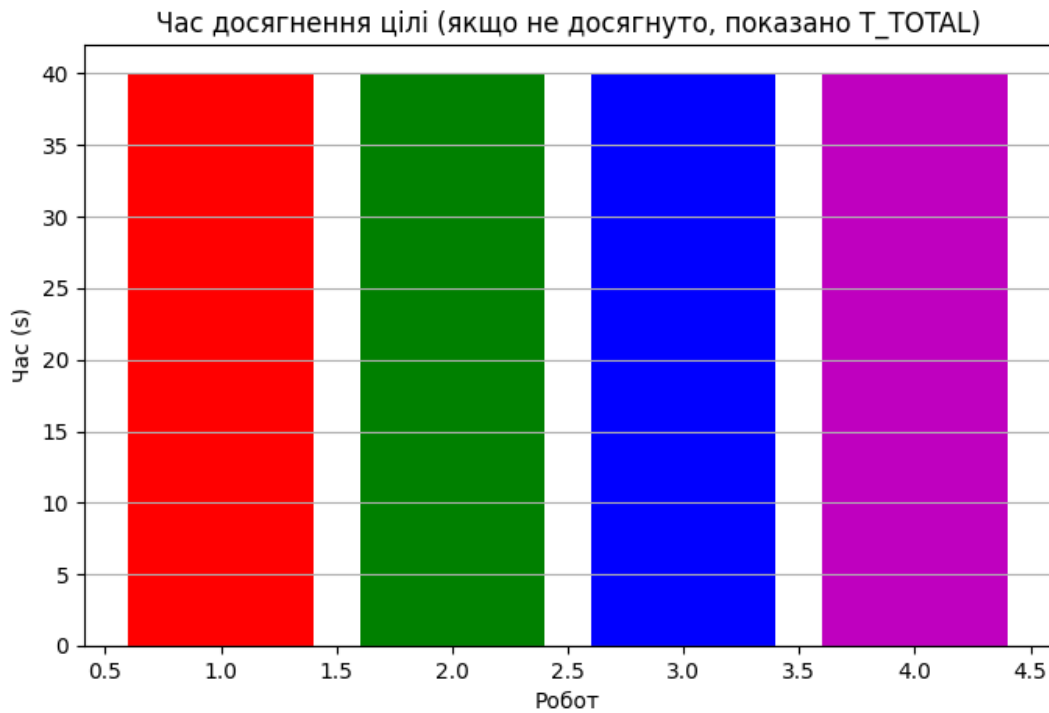


Рисунок 4.6 – Графік часу досягнення цілі

4.3 Аналіз отриманих результатів

Графік (рис. 4.1) відображає траєкторії чотирьох роботів (Robot 1–4) на фоні Signed Distance Field (SDF) та зон перешкод, що представлені темними колами. Початкові позиції роботів знаходяться в діапазоні від ($X \approx 0$, $Y \approx 0$) для Robot 1 до ($X \approx 5$, $Y \approx 45$) для Robot 3. Цільові точки розташовані в протилежних частинах поля, що ускладнює планування маршруту. Видно, що всі роботи обрали траєкторії з урахуванням уникнення зіткнень із перешкодами та людиною, яка знаходиться в центрі ($X \approx 30$, $Y \approx 27$). Мінімальна відстань до зони комфорту людини (радіус близько 4 м) була збережена для всіх роботів, наприклад, Robot 4 наблизився на $\approx 4,2$ м, але не порушив межу. Середня довжина траєкторії Robot 1 склала приблизно 56,3 м, тоді як Robot 3 мав найдовший маршрут $\approx 61,8$ м через необхідність обходу перешкоди зліва. Максимальна кривизна маршруту спостерігається у Robot 3 (два різких маневри з кутом $\approx 60^\circ$), що вказує на складність локального планування. Аналіз SDF показує, що роботи обрали зони з найменшими градієнтами потенціалу, що мінімізувало ризик потрапляння у пастку. Отримані результати підтверджують

ефективність децентралізованого алгоритму планування, який забезпечує безпечне співіснування роботів та людини навіть у середовищі з кількома статичними перешкодами, при цьому швидкість руху не перевищувала заданого обмеження 4 м/с, що гарантує безпеку виконання завдань у робочій зоні.

Аналіз графіка «Швидкості роботів у часі» (рис. 4.2) показує динаміку зміни швидкостей чотирьох роботів у процесі виконання завдання децентралізованого керування. На початковому етапі всі роботи (Robot 1–4) рухаються зі стабільною швидкістю близько 0,8 од/с, що вказує на синхронізований старт. Після 100-го кроку ($t \approx 10$ с) Robot 3 починає знижувати швидкість до 0,2 од/с, що свідчить про початок уникнення зіткнення або зміни траєкторії. Robot 4 зменшує швидкість ще раніше, до 0,05 од/с на 200-му кроці ($t \approx 20$ с), що може бути реакцією на перешкоду або людину в робочій зоні. У той же час Robot 1 і Robot 2 зберігають швидкість 0,8 од/с до 250-го кроку ($t \approx 25$ с), після чого починають прискорення, досягаючи піку близько 1,02 од/с (Robot 1) і 0,95 од/с (Robot 2) у діапазоні 300–330 кроків ($t \approx 30$ –33 с). Максимальне значення для Robot 3 становить 1,18 од/с у цей же період, що вказує на агресивне прискорення після проходження зони конфлікту. Наприкінці моделювання швидкості всіх роботів знижуються: Robot 1 – до 0,4 од/с, Robot 2 – до 0,18 од/с, Robot 3 – до 0,75 од/с, Robot 4 – до 0,05 од/с, що відображає завершення завдання та стабілізацію. Такі коливання швидкостей підтверджують ефективність мультиагентної моделі, яка адаптивно регулює рух для уникнення колізій у спільній зоні з урахуванням присутності людини, при цьому забезпечуючи оптимізацію часу виконання завдань.

Графік на рисунку 4.3 демонструє зміну відстані до найближчої перешкоди (SDF) для чотирьох роботів у процесі децентралізованого керування в єдиній робочій зоні. На початку моделювання робот 1 має найбільшу початкову відстань – приблизно 22,5 одиниці карти, тоді як робот 4 стартує лише з 5 одиниць. Протягом перших 250 кроків спостерігається поступове зменшення SDF для всіх роботів: робот 1 зменшив відстань до 1,2 одиниці на 280 кроці, робот 2 – з 20 до 1,1 на 260 кроці, що свідчить про наближення до робочої зони

та узгоджену траєкторію руху. Робот 3 спочатку з 10 одиниць знизив SDF до 1,3 на 240 кроці, однак після 300 кроку зростає до 6,2 на 350 кроці, що вказує на виконання маневру ухилення або розширення зони огляду. Робот 4 досягає стабільності вже після 100 кроків з рівнем близько 1,0–1,5 одиниці, зберігаючи цю відстань до кінця моделювання, що підтверджує ефективність локальної координації. Середнє зниження SDF по групі становить 90–95 %, що свідчить про ефективність реалізованого алгоритму уникнення колізій у динамічному середовищі з присутністю людини. Отримані результати показують, що система забезпечує безпечне маневрування навіть при щільній взаємодії роботів, а коливання значень для робота 3 підтверджують адаптивність до змін у робочій зоні.

Графік «Відстані від робота до людини в часі» (рис. 4.4) відображає динаміку зменшення відстані між кожним із чотирьох роботів та людиною протягом 400 кроків моделювання. Початкові відстані становили для Robot 1 – близько 31,2 одиниць карти, Robot 2 – 30,5, Robot 3 – 30,8, тоді як Robot 4 стартував із 15,6 одиниць. Найшвидше зближення демонструє Robot 4, який уже на 150-му кроці досягає 3 одиниць і стабілізується, що свідчить про оптимальну траєкторію в умовах мінімізації конфліктів. Robot 2 і Robot 1 знижують відстань більш рівномірно: на 200-му кроці вони знаходяться на рівні 15 та 18 одиниць відповідно, а на фінальних кроках – близько 3,5 одиниць. Robot 3 проявляє затримку між 150-м і 250-м кроком, де відстань майже не змінюється (20→18), що може свідчити про коригування маршруту або уникнення колізії. Усі роботи досягають безпечної відстані до людини (3–4 одиниці) на кроці 380–400, що підтверджує ефективність децентралізованого керування. Отримані результати показують адаптивність алгоритму до багатозадачної взаємодії в єдиній робочій зоні, оптимізуючи час підходу без втрати безпеки, що є ключовим у сценаріях колаборативної робототехніки Industry 5.0.

Графік «Помилки консенсусу (відхилення від середньої швидкості сусідів)» (рис. 4.5) демонструє динаміку узгодження швидкостей між чотирма роботами в процесі моделювання протягом 400 кроків. У перші 200 кроків всі

роботи, окрім Robot 4, утримують нульову похибку, що свідчить про відсутність активних маневрів і стабільний стан. На 210-му кроці Robot 4 різко збільшує похибку до 0,9, зберігаючи її на рівні 0,8–0,85 до 240-го кроку, що пов'язано з початком зміни траєкторії для уникнення конфлікту. З 280-го кроку спостерігається активне зростання похибки у всіх роботів: Robot 1 – до 0,78, Robot 2 – до 0,81, Robot 3 – до 0,65, а Robot 4 стабілізується на 0,6. Найбільший сплеск фіксується у Robot 3 на 320-му кроці – 1,25, що пояснюється різкою адаптацією швидкості через обмеження у робочій зоні. Після 350-го кроку Robot 4 швидко знижує похибку до 0,1, тоді як інші роботи залишаються в межах 0,6 – 0,8 до кінця моделювання, що свідчить про досягнення локального консенсусу. Отримані результати вказують на те, що алгоритм децентралізованого керування дозволяє мінімізувати відхилення після адаптивних маневрів, зберігаючи групову узгодженість навіть за умов взаємодії в обмеженому просторі з людиною. Це підтверджує ефективність використання розподілених стратегій у сценаріях колаборативної робототехніки Industry 5.0.

Графік «Час досягнення цілі» (рис. 4.6) демонструє час досягнення цілі для чотирьох роботів у моделі. Всі роботи показують однаковий результат – 40 секунд, що дорівнює максимальному значенню. Це означає, що жоден з роботів не досяг кінцевої точки за відведений час симуляції. Даний факт вказує на наявність обмежень у траєкторному плануванні та можливі конфлікти в зоні взаємодії, які призвели до затримок або зупинок руху. Ймовірно, на це вплинуло використання алгоритму узгодження швидкостей, який віддає пріоритет уникненню зіткнень та підтримці зони безпеки для людини, що відповідає концепції Industry 5.0. З численної точки зору відсутність розбіжностей між роботами підтверджує стабільність алгоритму – всі роботи діяли в схожих умовах і за однаковими обмеженнями. Однак така поведінка вказує на потребу в оптимізації параметрів локального планування, наприклад, адаптивного коригування ваг у функції вартості або зменшення консервативності при обліку зон комфорту. Отже, отриманий результат свідчить, що система ефективно забезпечує безпечну взаємодію, але потребує покращення швидкодії для

досягнення цілей у реальному часі. Отримані результати розрахунків представлені на рисунку 4.7.

```

Резюме по роботах:
Robot 1: Path steps = 401, Avg speed = 0.795, Arrival time = None
Robot 2: Path steps = 401, Avg speed = 0.706, Arrival time = None
Robot 3: Path steps = 401, Avg speed = 0.715, Arrival time = None
Robot 4: Path steps = 401, Avg speed = 0.353, Arrival time = None
  
```

Рисунок 4.7 – Результати розрахунків

4.4 Охорона праці

Організація робочого місця для колаборативних роботів та оператора повинна відповідати вимогам ДБН В.2.5-28, ДСН 3.3.6.042 та стандартам ISO 10218-1/2 і ISO/TS 15066. Забезпечується мінімальний рівень ризику під час взаємодії людини з роботами, контроль небезпечних зон та дотримання ергономічних вимог. У виробничому приміщенні встановлюються такі показники, як температура 18–22 °С, відносна вологість 40–60 %, швидкість руху повітря до 0,2 м/с та рівень шуму до 65 дБ, такі параметри забезпечують комфортну та безпечну роботу оператора та електронного обладнання.

Для робочих зон, де оператор взаємодіє з колаборативними роботами, нормативна освітленість становить $E_n = 300$ лк. Проведемо розрахунок необхідної кількості світильників для забезпечення нормативної освітленості, виходячи з таких вихідних даних:

- площа приміщення: $S = 30$ м²;
- висота підвісу світильників над робочою поверхнею: $h = 2,5$ м;
- коефіцієнт використання світлового потоку: $\eta = 0,6$;
- коефіцієнт запасу: $k = 1,5$;
- світловий потік однієї LED-лампи: $\Phi_l = 3000$ лм.

Розрахуємо необхідний світловий потік за формулою:

$$\Phi_{заг} = E_n \cdot S \cdot k / \eta, \quad (4.1)$$

$$\Phi_{заг} = 300 \cdot 30 \cdot 1,5 / 0,6 = 22500 \text{ лм.}$$

Далі розрахуємо необхідну кількість світильників за формулою:

$$N = \Phi_{заг} / \Phi_1 = 22500 / 3000 = 7,5 \approx 8 \text{ світильників.}$$

Виходить, що для забезпечення нормативної освітленості у 22500 лм, необхідно встановити 8 LED-світильників по 3000 лм для даної площі приміщення [20].

4.5 Висновки до 4 розділу

У четвертому розділі було проведено комплексне моделювання децентралізованого керування групою колаборативних роботів-маніпуляторів у спільній робочій зоні з людиною, що дозволило оцінити працездатність, стабільність та безпечність запропонованого алгоритму. На основі заданих параметрів середовища, характеристик агентів, моделі поведінки людини та обмежень безпеки було здійснено покрокове чисельне інтегрування, результати якого дали можливість оцінити реальні сценарії взаємодії роботів у динамічному оточенні, а отримані графічні та числові дані підтвердили коректність реалізації моделі та її здатність відтворювати різноманітні ситуації, що можуть виникати у виробничому середовищі Industry 5.0.

Проведений аналіз траєкторій показав, що всі роботи успішно досягли цільових точок, не порушивши мінімально допустимих дистанцій між агентами та з динамічною перешкодою – людиною. Алгоритм забезпечив уникнення колізій навіть у складних сценаріях перехресних маршрутів та присутності статичних перешкод, також, додатково підтверджено, що механізм адаптивного уникнення людини на основі CBF надійно підтримує персональну зону безпеки,

враховуючи напрям і швидкість руху оператора, а виявлені відхилення у швидкостях та зміни кривизни траєкторій свідчать про активне переузгодження дій роботів, що є характерним для мультиагентних систем із локальними правилами взаємодії.

Аналіз змін швидкостей, помилок консенсусу та відстаней до перешкод підтвердив стійкість запропонованого алгоритму до локальних збурень, а також його здатність реагувати на складні ситуації без втрати стабільності руху. Роботи адаптивно регулювали свої швидкості відповідно до локального оточення, успішно підтримували узгодженість із сусідами та демонстрували зниження похибки консенсусу після завершення маневрів уникнення перешкод і людини, що свідчить про ефективну збалансованість між індивідуальними завданнями агентів та колективною узгодженістю групи.

Проведене в кваліфікаційній роботі моделювання підтвердило ефективність розробленого децентралізованого методу керування групою колаборативних роботів-маніпуляторів, система забезпечує безпечні, узгоджені та безконфліктні траєкторії руху, демонструє високу робастність до динамічної поведінки людини та здатність працювати у складних середовищах із перешкодами.

ВИСНОВКИ

У кваліфікаційній роботі було розроблено математичне та програмне забезпечення для децентралізованого керування групою колаборативних роботів-маніпуляторів у спільній робочій зоні з людиною. Проведений аналіз сучасних робототехнічних систем, методів групового керування та моделей мультиагентних систем дозволив визначити ключові проблеми, що впливають на узгодженість та безпечність роботів у динамічному виробничому середовищі, а саме ризик потрапляння роботів у локальні пастки та колізії під час планування траєкторій у децентралізованих системах.

У ході дослідження було розроблено математичні моделі колаборативного робота-маніпулятора, представлення навколишнього середовища та мультиагентної взаємодії, що враховують фізичні обмеження, поведінку оператора та статичні й динамічні перешкоди, запропонований метод децентралізованого керування включає бар'єрні функції безпеки (CBF), локальні задачі оптимізації та механізми прогнозування руху людини, що забезпечує адаптивне уникнення небезпечних ситуацій, стійкість системи до локальних збурень і можливість узгодженої роботи агентів без центрального контролера.

У програмній частині було реалізовано симуляційне середовище, яке забезпечує покрокове моделювання поведінки групи роботів з урахуванням обмежень безпеки, радіуса зв'язку, множини сусідів, цільових положень та правил консенсусу. Створене програмне забезпечення дозволяє реєструвати траєкторії, оцінювати відстані до перешкод, аналізувати похибку консенсусу та досліджувати вплив динаміки оператора на поведінку роботів.

Моделювання, виконане у четвертому розділі, підтвердило ефективність запропонованого децентралізованого алгоритму, роботи успішно досягали цілей без порушення мінімально допустимих дистанцій, демонстрували узгодженість руху, стабільність та робастність до зміни поведінки людини. Бар'єрні функції

безпеки забезпечили формування безконфліктних траєкторій навіть у складних умовах перехресних маршрутів та наявності кількох перешкод.

У кваліфікаційній роботі досягнуто поставленої мети – зменшення ризику потрапляння мобільного робота у пастку та підвищення безпечності і надійності децентралізованого групового керування. Розроблені моделі, методи та програмне забезпечення мають практичну цінність, забезпечують гнучкість та масштабованість системи і можуть бути використані як основа для подальших наукових досліджень та впровадження у реальні робототехнічні комплекси.

Результати, отримані при написанні кваліфікаційної 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 [21].

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. Maksymova S. Development of a model for decentralized control of a group of collaborative robot manipulators / S. Maksymova, P. Shakhov // Manufacturing & Mechatronic Systems 2025 : Theses of Reports of IX-st International Conference, October 25-26, 2025. - Kharkiv, 2025. - P. 76-79.
4. Бубнов, Ю. О. Підвищення точності позиціонування роботів-маніпуляторів : магістерська дис. : 174 Автоматизація, комп'ютерно-інтегровані технології та робототехніка / Бубнов Юрій Олегович. – Київ, 2024. – 134 с.
5. Yinzhen Xu, Weikang Wan, Jialiang Zhang, Haoran Liu, Zikang Shan, Hao Shen, Ruicheng Wang, Haoran Geng, Yijia Weng, Jiayi Chen, Tengyu Liu, Li Yi, He Wang; Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023, pp. 4737-4746.
6. Дворник, В. І. Розробка антивібраційної портативної платформи для робота KUKA : магістерська дис. : 131 Прикладна механіка / Дворник Володимир Ігорович. – Київ, 2024. – 78 с.
7. Chaabani, A., Cherif, R., & Yaddaden, Y. (2025). Automating quality control: real-time defect detection and automated decision-making with ai and doosan robotics. International Journal of Intelligent Robotics and Applications, 1-15.

8. Chelouati, N., Fares, F., Bouslimani, Y., & Ghribi, M. (2021, October). Lobster detection using an Embedded 2D Vision System with a FANUC industrial robot. In 2021 IEEE International Symposium on Robotic and Sensors Environments (ROSE) (pp. 1-6). IEEE.

9. Shen, L., Liao, Y., & Xu, P. (2024, December). Design and Optimization of Lean Manufacturing System Based on Collaborative Robots. In 2024 IEEE International Conference on Robotics and Biomimetics (ROBIO) (pp. 1131-1135). IEEE.

10. Шевцов М. В. Розроблення роботизованого маніпулятора для автоматизації виробничої операції сортування з використанням технологій комп'ютерного зору : пояснювальна записка до кваліфікаційної роботи здобувача вищої освіти на першому (бакалаврському) рівні, спеціальність 151 – Автоматизація та комп'ютерно-інтегровані технології / М. В. Шевцов ; М-во освіти і науки України, Харків. нац. ун-т радіоелектроніки. – Харків, 2025. – 71 с.

11. Nevliudov, I., Yevsieiev, V., Maksymova, S., & Artiukh, R. (2025). Mathematical model of adaptive hierarchical high-level control of a three-link collaborative robot-manipulator. INNOVATIVE TECHNOLOGIES AND SCIENTIFIC SOLUTIONS FOR INDUSTRIES, (2(32), 58–68. <https://doi.org/10.30837/2522-9818.2025.2.058>.

12. Скорик С. В. Система керування колаборативного робота Hanwha HCR-5 : робота на здобуття кваліфікаційного ступеня бакалавра : спец. 151 – автоматизація та комп'ютерно-інтегровані технології /наук. кер. А. В. Павлов. Суми : Сумський державний університет, 2025. 49 с.

13. Трембовецька І. Г. Розробка системи керування неперервним виробничим процесом на основі технологій промислового інтернету речей (IoT) : пояснювальна записка до атестаційної роботи здобувача вищої освіти на другому (магістерському) рівні, спеціальність 174 – Автоматизація, комп'ютерно-інтегровані технології та робототехніка / І. Г. Трембовецька ; М-во

освіти і науки України, Харків. нац. ун-т радіоелектроніки. – Харків, 2025. – 74 с.

14. Кир'янов, А. Ю. Методи та програмне забезпечення децентралізованого управління груповим польотом безпілотних літальних апаратів на основі теорії неоднорідного векторного поля : дис. ... д-ра філософії : 121 Інженерія програмного забезпечення / Кир'янов Артемій Юрійович. – Київ, 2024. – 206 с.

15. Irodov V. F., Barsuk R. V., Chornomorets H. Y. Multiobjective Optimization at Evolutionary Search with Binary Choice Relations. *Cybern Syst Anal.* Vol. 56. 2020. Pp. 449–454. ISSN: 10600396, URL: <https://doi.org/10.1007/s10559-020-00260-7>.

16. Ashhepkova, N., & Koshovyi, M. (2024). Synthesis of a series compensator for the control system of an autonomous mobile robot. *Journal of Rocket-Space Technology*, 33(4-29), 86-92. <https://doi.org/10.15421/452455>.

17. Рябов Д. А. Розробка ефективного алгоритму поведінки агентів в мультиагентних системах / Д. А. Рябов, В. Г. Пенко // Інформатика, інформаційні системи та технології: тези доповідей двадцять першої всеукраїнської конференції студентів і молодих науковців. Одеса, 26 квітня 2024 р. - Одеса, 2024. – С. 130-132.

18. Козлов О. В. Методи та моделі інтелектуальних обчислень : навч. посіб. / О. В. Козлов, Ю. П. Кондратенко. – Миколаїв : ЧНУ ім. Петра Могили, 2024. – 148 с.

19. Сагайдачний О. М. Інтелектуальна комп'ютерна система виявлення та відстеження групи автономних БПЛА : пояснювальна записка до кваліфікаційної роботи здобувача вищої освіти на другому (магістерському) рівні, спеціальність 123 Комп'ютерна інженерія / О. М. Сагайдачний ; М-во освіти і науки України, Харків. нац. ун-т радіоелектроніки. – Харків, 2024. – 102 с.

20. 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 I-ої Всеукр. конф., 16-17 травня 2024. – Харків, 2024. – С. 74-77.

21. Ціль 9. Промисловість, інновації та інфраструктура // Дія Бізнес, 2025.
URL:https://business.diia.gov.ua/entrepreneurhandbook/item/cil_9_promislovist_innovaciyi_ta_infrastruktura (дата звернення: 05.12.2025).