

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Автоматики і комп'ютеризованих технологій
(повна назва)

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

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий
(магістерський) Розроблення програмного модулю для управління
роботизованою платформою з використанням технології ПОТ
(тема)

Виконав:
здобувач 2 року навчання,
групи КТРСм-23-1
Ісмайлов Т.В.
(прізвище, ініціали)

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

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

Освітня програма Комп'ютеризовані та
робототехнічні системи
(повна назва освітньої програми)

Керівник доц.Стародубцев М. Г.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри

(підпис)

(прізвище, ініціали)

2025 р.

Харківський національний університет радіоелектроніки

Факультет Автоматики і комп'ютеризованих технологійКафедра Комп'ютерно-інтегрованих технологій, автоматизації та
робототехнікиРівень вищої освіти другий (магістерський)Спеціальність 174 Автоматизація, комп'ютерно-інтегровані технології та
робототехніка

(код і повна назва)

Тип програми освітньо професійна

(освітньо-професійна або освітньо-наукова)

Освітня програма Комп'ютеризовані та робототехнічні системи
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

« ____ » _____ 20 ____ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Ісмайлову Тимуру Віладійовичу
(прізвище, ім'я, по батькові)1. Тема роботи: Розроблення програмного модулю для управління
роботизованою використанням технології ПОТ

затверджена наказом університету від 25 "листопада" 2024 р. № 1239

2. Термін подання здобувачем роботи до екзаменаційної комісії 23 січня 2025 р.3. Вихідні дані до роботи роботизована платформа; алгоритми візуалізації руху
роботизованих платформ; IBM-сумісний комп'ютер; Robotino Sim Demo,
Robotino View.4. Перелік пунктів що потрібно опрацювати в роботі: вступ; огляд сучасних
алгоритмів візуалізації руху, їх класифікація та аналіз; опис особливостей роботи
алгоритмів; аналіз вимог до реалізації алгоритмів; постановка мети і завдань;
побудова математичних моделей та алгоритмів; розробка тестових сценаріїв та

експериментальна перевірка; порівняльний аналіз ефективності алгоритмів у реальних умовах; 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 PowerPoint (*.ppt) – 16 с. формату А4

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

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

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на виконання роботи	25.11.24	Виконано
2	Аналіз предметної області	27.11.24	Виконано
3	Постановка мети і задач кваліфікаційної роботи	29.11.24	Виконано
4	Вибір методів розв'язання задачі	3.12.24	Виконано
5	Розробка програмного забезпечення задачі інформаційної системи	8.12.24	Виконано
6	Оформлення пояснювальної записки	17.12.24	Виконано
7	Подання закінченої роботи керівникові	30.12.24	Виконано
8	Усунення зауважень керівника	04.01.25	Виконано
9	Підготовка презентації	06.01.25	Виконано
10	Подання роботи на рецензування	9.01.25	Виконано

Дата видачі завдання 25 листопада 2024 р.

Здобувач 
(підпис)

Керівник роботи _____

доц. Стародубцев М.Г

(підпис)

(посада, прізвище, ініціали)

Я, як студент ХНУРЕ, розумію і підтримую політику закладу із академічної доброчесності. Я не надавав і не одержував недозволену допомогу під час підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

«25» листопада 2024 р.



Ісмайлов Т.В.

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи: 90 с., 1 табл., 23 рис., 3 дод., 20 джерел.

РОБОТОТЕХНІКА, АЛГОРИТМИ ВІЗУАЛІЗАЦІЇ РУХУ, ПРОГРАМУВАННЯ, СИМУЛЯЦІЙНЕ СЕРЕДОВИЩЕ.

Мета роботи – підвищення ефективності роботи роботизованих платформ за рахунок впровадження системи візуалізації, яка забезпечує точне розпізнавання перешкод, визначення оптимальних траєкторій руху та адаптацію до змінних умов середовища.

Об’єкт дослідження – процес управління роботизованою платформою в симуляційному середовищі.

Предмет дослідження – алгоритми візуалізації руху для роботизованих платформ.

Методи вирішення завдання – симуляційне моделювання, методи обробки сенсорних даних, математичне моделювання, кінематичні та динамічні методи, алгоритми сенсорного злиття, системний аналіз.

Результати – розроблено компоненти предметної області «Розробка алгоритмів візуалізації руху»: сформульовано мету та завдання кваліфікаційної роботи; обрано інструменти для моделювання та програмування; протестовано алгоритми у симуляційному середовищі Robotino SIM Demo; виконано інтеграцію та тестування алгоритмів на реальній платформі.

Напрямами подальших досліджень можуть бути:

- впровадження методів машинного навчання для покращення адаптивності платформ;
- дослідження оптимізації алгоритмів для роботи в реальному часі;

– розробка нових підходів до інтеграції з сенсорами та зовнішніми системами.

Також, отриманні результати роботи можна віднести до Цілі сталого розвитку 9 – “Промисловість, інновації та інфраструктура” а саме п. 9.1: “Розвивати якісну, надійну, сталу та доступну інфраструктуру” Роботизована платформа, розроблена у рамках цієї роботи оптимізує інфраструктуру у багатьох напрямках, таких як: промисловість, медицина, логістика та інші. Завдяки інноваційній технології роботизована платформа підвищує ефективність виробництва та відповідає вимогам сталого розвитку.

ABSTRACT

Explanatory note to the bachelor's qualification thesis: 90 p., 1 table, 23 figures, 3 app., 20 sources.

ROBOTICS, MOTION VISUALIZATION ALGORITHMS,
PROGRAMMING, SIMULATION ENVIRONMENT.

The purpose of the work is to increase the efficiency of robotic platforms by implementing a visualization system that provides accurate obstacle recognition, determination of optimal movement trajectories, and adaptation to changing environmental conditions.

The object of research is the process of controlling a robotic platform in a simulation environment.

The subject of the study is motion visualization algorithms for robotic platforms.

Methods for solving the problem are simulation modeling, sensor data processing methods, mathematical modeling, kinematic and dynamic methods, sensor fusion algorithms and system analysis.

Results – components of the subject area "Development of motion visualization algorithms" have been developed: the goal and objectives of the qualification work have been formulated; tools for modeling and programming have been selected; algorithms have been tested in the Robotino SIM Demo simulation environment; integration and testing of algorithms on a real platform have been performed:

- further research directions may include: implementing machine learning methods to improve platform adaptability;
- research into optimizing algorithms for real-time operation;

- development of new approaches to integration with sensors and external systems.

Also, the results of the work can be attributed to Sustainable Development Goal 9 – “Industry, Innovation and Infrastructure” namely, item 9.1: “Develop high-quality, reliable, sustainable and accessible infrastructure” The robotic platform developed within the framework of this work optimizes infrastructure in many areas, such as: industry, medicine, logistics and others. Thanks to innovative technology, the robotic platform increases production efficiency and meets the requirements of sustainable development.

ЗМІСТ

Перелік скорочень.....	11
Вступ.....	12
1 Огляд алгоритмів візуалізації руху роботизованих платформ.....	14
1.1 Алгоритми візуалізації руху роботизованих платформ: важливість і загальний огляд.....	14
1.2 Алгоритми візуалізації в різних галузях.....	16
1.3 Висновки до розділу.....	22
2 Підготовка до розробки програмного забезпечення.....	24
2.1 Програмні засоби для моделювання та тестування роботизованих платформ.....	24
2.2 Детальна класифікація алгоритмів візуалізації руху.....	25
2.3 Проблеми та виклики в реалізації алгоритмів візуалізації.....	42
2.4 Порівняльний аналіз ефективності різних алгоритмів в умовах динамічного середовища.....	44
2.5 Порівняльна таблиця.....	49
2.6 Технологічні перспективи та можливості розвитку.....	52
2.7 Проблеми та виклики в реалізації алгоритмів візуалізації.....	55
2.8 Висновки до розділу.....	59
3 Побудова алгоритмів роботи програмного забезпечення.....	60
3.1 Основні поняття та визначення.....	60
3.2 Моделювання алгоритмів у Robotino SIM Demo.....	60
3.3 Інтеграція алгоритмів у Robotino View 2.....	61
3.4 Висновки до розділу.....	61
4 Опис реалізації програмного забезпечення.....	63
4.1 Середовище розробки проекту.....	63
4.2 Реалізація алгоритмів.....	63
4.3 Порядок розробки програми для уникнення перешкод мобільним роботом.....	64
4.4 Висновки до розділу.....	69
5 Охорона праці та безпека у надзвичайних ситуаціях.....	71
5.1 Промислова безпека в середовищі робототехнічних досліджень.....	71
5.2 Виробнича санітарія у лабораторії робототехніки.....	72
5.3 Безпека у надзвичайних ситуаціях.....	72
5.4 Висновки до розділу.....	73
Висновки.....	75
Перелік джерел посилання.....	77

Додаток А Апробація наукових результатів.....	80
Додаток Б Демонстраційний матеріал.....	89

ПЕРЕЛІК СКОРОЧЕНЬ

ШІ – штучний інтелект;

AR– доповнена реальність;

VR – віртуальна реальність.

ВСТУП

На сучасному етапі розвитку робототехніки спостерігається значне зростання популярності автономних платформ, які використовуються у промисловості, медицині, логістиці, освіті та інших галузях. Ці платформи забезпечують автоматизацію складних процесів, підвищуючи продуктивність і знижуючи потребу в ручному управлінні. Однак, успішне функціонування таких систем потребує ефективних алгоритмів візуалізації та управління рухом, які дозволяють платформам адаптуватися до динамічних умов середовища.

Сучасний стан проблеми вказує на необхідність розробки нових алгоритмів та вдосконалення існуючих підходів до візуалізації руху. Протягом останніх років досягнуто значного прогресу у цій сфері, але існують невирішені питання, пов'язані з точністю алгоритмів, інтеграцією сенсорних систем та енергоефективністю. Ці аспекти залишаються важливими напрямками для досліджень.

У світовому контексті вирішення цих задач стало пріоритетом для багатьох компаній та дослідницьких центрів. Розробка інноваційних алгоритмів для роботизованих платформ активно впроваджується у транспорті, виробничих процесах, системах автономного управління та інших галузях. Такі дослідження є актуальними та мають значний вплив на розвиток технологій автоматизації.

З огляду на це, дослідження та розробка алгоритмів візуалізації руху для роботизованих платформ є актуальними завданнями, які сприяють підвищенню точності, автономності та ефективності роботів. Результати цієї роботи дозволять покращити функціональні можливості платформ та забезпечити їх стабільну роботу в умовах змінного середовища.

Об'єкт дослідження – роботизована платформа, що слугує технічним засобом для виконання завдань у реальному середовищі.

Предметом дослідження є алгоритми візуалізації руху роботизованих платформ.

Мета роботи – підвищення ефективності роботи роботизованих платформ за рахунок впровадження системи візуалізації, яка забезпечує точне розпізнавання перешкод, визначення оптимальних траєкторій руху та адаптацію до змінних умов середовища.

Для досягнення поставленої мети було сформульовано наступні завдання:

- а) виконати аналіз існуючих алгоритмів візуалізації руху;
- б) здійснити вибір інструментів для моделювання та реалізації алгоритмів;
- в) розробити алгоритми візуалізації для конкретних сценаріїв руху;
- г) протестувати алгоритми у симуляційному середовищі;
- д) адаптувати алгоритми для роботи на реальній платформі.

Роботу виконано згідно з вимогами ДСТУ 3008:2015 [1].

Також на основі дипломної роботи була опублікована публікація [2].

1 ОГЛЯД АЛГОРИТМІВ ВІЗУАЛІЗАЦІЇ РУХУ РОБОТИЗОВАНИХ ПЛАТФОРМ

1.1 Алгоритми візуалізації руху роботизованих платформ: важливість і загальний огляд

Алгоритми візуалізації руху роботизованих платформ є ключовим елементом сучасних автоматизованих систем, оскільки вони забезпечують оптимальні рішення для контролю траєкторії руху в реальних умовах. Їх основна функція полягає в забезпеченні точного керування рухом роботів, що дозволяє ефективно виконувати складні завдання з високою точністю та швидкістю. Ці алгоритми сприяють підвищенню автономності роботизованих платформ, що є критично важливим для промислових процесів, де необхідно забезпечити високу продуктивність при мінімальному втручанні з боку людини. Завдяки алгоритмам візуалізації системи можуть адаптуватися до змінних умов навколишнього середовища, що знижує ризик виникнення помилок та підвищує рівень безпеки. Крім того, алгоритми візуалізації відіграють значну роль у поліпшенні взаємодії між людиною та машиною, сприяючи збільшенню рівня автоматизації, що дозволяє скоротити потребу в постійному людському контролі та підвищити ефективність виробничих процесів.

Розвиток цих технологій має значний вплив на ефективність роботизованих систем у різних галузях. Наприклад, у медицині алгоритми візуалізації забезпечують точне керування хірургічними роботами під час проведення складних операцій, де необхідна надзвичайна точність у виконанні мікро-маніпуляцій. У промисловості роботизовані системи можуть автономно контролювати свої дії, що дозволяє оптимізувати виробничі процеси, зменшити час простоїв та знизити витрати на обслуговування. У транспортних системах, зокрема автономних автомобілях, алгоритми візуалізації використовуються для орієнтації на дорозі, швидкого

розпізнавання перешкод та уникнення зіткнень, що підвищує загальний рівень безпеки на дорогах. У військових технологіях роботизовані платформи (рис. 1.1), оснащені такими алгоритмами, здатні виконувати завдання в умовах підвищеного ризику або в складній місцевості, де необхідна швидка адаптація до змінних умов [3], [4], [5].

Загалом, розвиток алгоритмів візуалізації руху роботизованих систем має суттєвий вплив на робототехніку та автоматизацію, відкриваючи нові перспективи для їх застосування в таких галузях, як наука, транспорт, медицина, військові технології та розваги.

Ці алгоритми дозволяють роботам працювати з підвищеною точністю та автономією, що не лише знижує ризики для людей, але й значно підвищує продуктивність.

Удосконалення цих технологій робить роботизовані системи більш адаптивними, надійними та ефективними, що сприяє їх широкому впровадженню в різноманітних сферах.

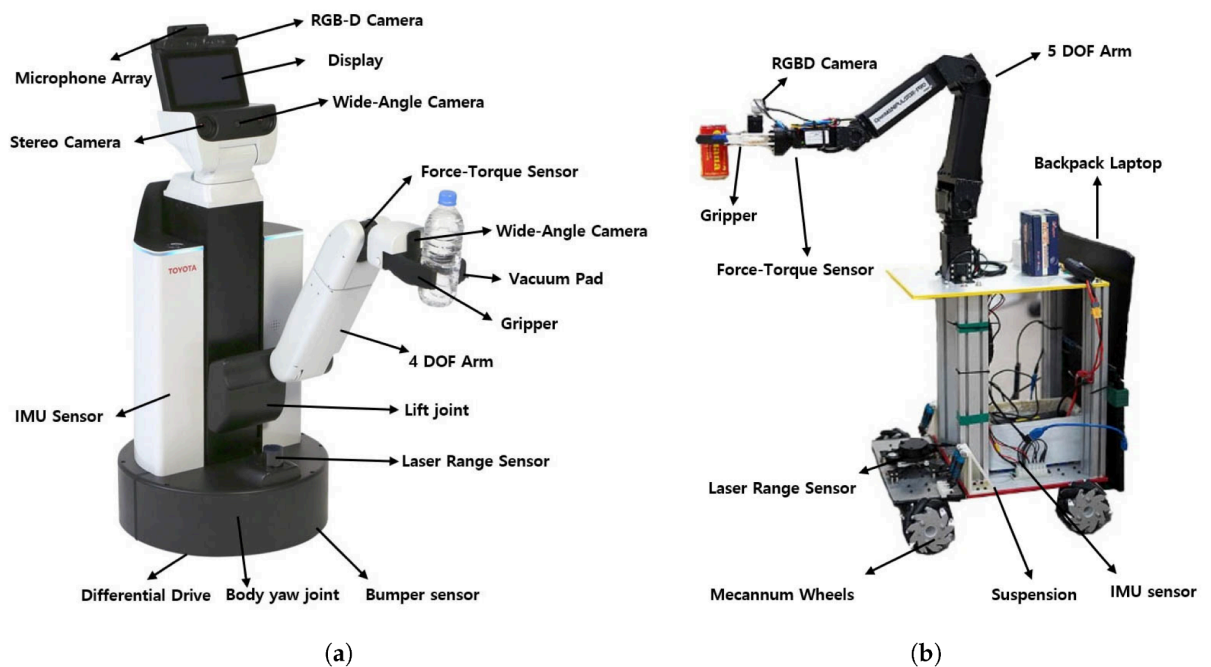


Рисунок 1.1 – Приклад роботизованої платформи [5]

1.2 Алгоритми візуалізації в різних галузях

Алгоритми візуалізації руху знаходять широке застосування в багатьох галузях, де точність і автоматизація є критичними для досягнення високої ефективності та безпеки процесів. Вони дозволяють як роботам, так і операторам контролювати динамічні середовища, вчасно коригувати рухи, приймати рішення на основі отриманих даних та адаптуватися до непередбачуваних змін. Ці алгоритми інтегровані у різноманітні автоматизовані системи, забезпечуючи ефективне функціонування роботизованих платформ у складних і мінливих умовах. У залежності від сфери застосування, алгоритми можуть бути налаштовані на виконання специфічних завдань, таких як зварювання, хірургія або логістика, що відкриває перед роботизованими системами широкі можливості для впровадження у сучасні технологічні процеси.

1.2.1 Промисловість

У виробничих процесах (рис. 1.2), особливо в таких точних сферах, як зварювання або обробка матеріалів, алгоритми візуалізації відіграють важливу роль у забезпеченні точності рухів роботів [4], [6]. Візуалізація дозволяє операторам та автоматизованим системам спостерігати за кожним етапом циклу виробництва і вчасно вносити корективи, що підвищує ефективність виробничого процесу [7], [8]. Наприклад, зварювальні роботи повинні виконувати рівномірний розподіл зусиль на матеріалах, і кінематичні алгоритми допомагають знизити кількість помилок, забезпечуючи стабільність та високу якість роботи. Завдяки алгоритмам візуалізації можна досягти значного підвищення точності у виконанні промислових завдань. В промисловості використовуються такі алгоритми візуалізації руху: алгоритми на основі карт і SLAM (Simultaneous Localization and Mapping), алгоритми планування траєкторії, алгоритми на основі штучного інтелекту, алгоритми

контролю руху (Motion Control Algorithms), алгоритми обробки зображень та розпізнавання об'єктів, алгоритми на основі графічних рушіїв.



Рисунок 1.2 – Роботи у виробничих процесах [6]

1.2.2 Медицина

У галузі медицини алгоритми візуалізації є важливими для точного керування хірургічними роботами, такими як робот Da Vinci (рис. 1.3) [4], [9]. Такі алгоритми дозволяють хірургам не тільки спостерігати за поточним положенням інструментів, але й прогнозувати їхні майбутні рухи, що суттєво знижує ризики під час складних хірургічних операцій. Можливість передбачити наступні дії робота допомагає мінімізувати помилки та покращити загальний результат хірургічного втручання. Точність та надійність, яку забезпечують ці алгоритми, робить їх незамінними у сфері роботизованої медицини, де навіть найменша неточність може мати критичні наслідки.

В медицині використовуються такі алгоритми візуалізації руху: алгоритми обробки зображень, алгоритми машинного навчання та штучного інтелекту, алгоритми планування траєкторії, алгоритми контролю руху

(Motion Control), алгоритми навігації, алгоритми симультанної локалізації та побудови карти (SLAM), алгоритми стеження та прогнозування руху, алгоритми віртуальної та доповненої реальності.

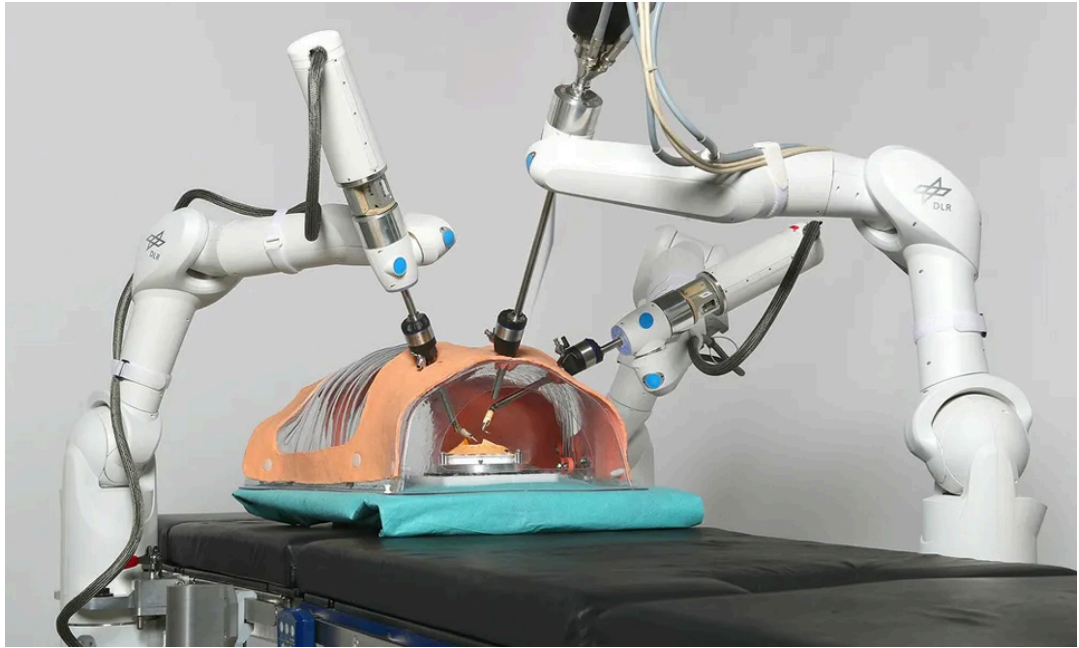


Рисунок 1.3 – Роботи у медицині [9]

1.2.3 Логістика

У логістичних центрах, де роботи (рис. 1.4) використовуються для транспортування товарів, алгоритми візуалізації відіграють важливу роль у координуванні руху роботів [4], [10]. Вони дозволяють автоматизованим платформам прокладати оптимальні маршрути, уникати зіткнень і ефективно доставляти товари. Завдяки цим алгоритмам логістичні процеси значно прискорюються, зменшуються витрати на транспортування та підвищується загальна ефективність роботи. Це має велике значення для складів та логістичних компаній, які повинні обробляти величезні обсяги вантажів і забезпечувати їх швидку доставку [11].

В логістиці використовуються такі алгоритми візуалізації руху: алгоритми оптимізації маршрутів (Route Optimization Algorithms), алгоритми планування транспортування (Transportation Planning Algorithms), алгоритми

управління складом (Warehouse Management Algorithms), алгоритми навігації (Navigation Algorithms), алгоритми управління запасами (Inventory Management Algorithms), алгоритми планування робочих процесів (Workflow Planning Algorithms), алгоритми прогнозування попиту (Demand Forecasting Algorithms), алгоритми збирання даних (Data Collection Algorithms), алгоритми симуляції та моделювання (Simulation and Modeling Algorithms), алгоритми управління транспортними засобами (Fleet Management Algorithms) [12], [13].



Рисунок 1.4 – Роботи у логістичних центрах [10]

1.2.4 Автономні транспортні засоби

Алгоритми візуалізації руху є основою для роботи автономних транспортних засобів (рис. 1.5), таких як безпілотні автомобілі [4], [14]. Вони дозволяють транспортним засобам аналізувати оточення у реальному часі, обирати оптимальні маршрути та уникати перешкод. Системи, що базуються на LIDAR та комп'ютерному зорі, надають автомобілям точні дані про їхнє середовище, що дозволяє приймати рішення в режимі реального часу. Це підвищує безпеку на дорогах і робить можливим широке впровадження

автономних транспортних засобів у реальне життя. Алгоритми дозволяють транспортним системам адаптуватися до постійно змінюваних умов на дорогах і забезпечують надійність роботи в різних середовищах.

Для автономних транспортних засобів використовуються такі алгоритми візуалізації руху: алгоритми комп'ютерного зору (Computer Vision Algorithms), алгоритми виявлення та уникнення перешкод (Obstacle Detection and Avoidance Algorithms), алгоритми планування траєкторії (Trajectory Planning Algorithms), алгоритми симультанної локалізації та побудови карти (SLAM), алгоритми навігації на основі GPS (GPS-based Navigation Algorithms), алгоритми об'єднання сенсорних даних (Sensor Fusion Algorithms), алгоритми прогнозування поведінки (Behavior Prediction Algorithms), алгоритми контролю руху (Motion Control Algorithms), алгоритми ухвалення рішень (Decision-Making Algorithms), алгоритми машинного навчання та штучного інтелекту (Machine Learning and AI Algorithms), алгоритми комунікації між транспортними засобами (V2V/V2I Communication Algorithms), алгоритми кібербезпеки (Cybersecurity Algorithms) [7], [8].



Рисунок 1.5 – Автономний транспорт (Tesla Model S) [14]

1.2.5 Наукові дослідження

Роботизовані платформи активно використовуються в наукових дослідженнях, зокрема у вивченні океанів та космосу, де людське втручання є обмеженим або неможливим.

Алгоритми візуалізації дозволяють вченим контролювати рух роботів (рис. 1.6) у складних середовищах, що відкриває нові можливості для проведення наукових експериментів та досліджень [4], [15].

Завдяки таким алгоритмам дослідники можуть отримувати точні дані про середовище, в якому працюють роботи, мінімізуючи ризики для самих вчених.

Це сприяє важливим відкриттям у таких галузях, як геологія, океанологія та астрономія.

Загалом, алгоритми візуалізації руху роботизованих платформ відіграють ключову роль у багатьох галузях, забезпечуючи точність, безпеку та ефективність у виконанні різноманітних завдань. Завдяки постійному розвитку цих технологій, роботи стають більш автономними і здатними виконувати все складніші завдання у мінливих умовах, що відкриває нові перспективи для їх використання у найрізноманітніших сферах.

Для наукових досліджень використовуються такі алгоритми візуалізації руху:

алгоритми симуляції руху (Motion Simulation Algorithms), алгоритми зворотної кінематики (Inverse Kinematics Algorithms), алгоритми комп'ютерного зору (Computer Vision Algorithms), алгоритми машинного навчання (Machine Learning Algorithms), алгоритми контролю руху (Motion Control Algorithms), алгоритми навігації (Navigation Algorithms), алгоритми SLAM (Simultaneous Localization and Mapping), алгоритми для збору та обробки даних (Data Acquisition and Processing Algorithms), алгоритми оптимізації (Optimization Algorithms), алгоритми симуляції фізичних процесів (Physical Process Simulation Algorithms), алгоритми розподілених систем

(Distributed Systems Algorithms), алгоритми візуалізації великих даних (Big Data Visualization Algorithms)



Рисунок 1.6 – Робот Pepper [15]

1.3 Висновки до розділу

У розділі було проведено огляд алгоритмів візуалізації руху роботизованих платформ, визначено їхню важливість у різних сферах застосування, таких як промисловість, медицина, логістика, автономний транспорт та наукові дослідження.

Алгоритми візуалізації руху відіграють ключову роль у забезпеченні точності, стабільності та адаптивності роботизованих систем. Вони дозволяють ефективно керувати платформами в умовах змінного середовища, знижуючи ризик помилок і підвищуючи продуктивність.

- у промисловості алгоритми забезпечують автоматизацію та точність виконання технологічних процесів, таких як зварювання та обробка матеріалів. Це дозволяє значно підвищити ефективність виробництва;

- у медицині алгоритми візуалізації забезпечують надзвичайну точність роботи хірургічних роботів, що суттєво знижує ризики під час складних операцій і підвищує якість медичних втручань;

- у логістиці вони оптимізують транспортні процеси, забезпечуючи ефективну роботу складів та автоматизованих транспортних систем, що сприяє підвищенню швидкості та точності доставки вантажів;

- у сфері автономного транспорту алгоритми візуалізації дозволяють автомобілям аналізувати дорожню ситуацію, уникати зіткнень та адаптуватися до змінних умов, забезпечуючи безпеку та надійність;

- у наукових дослідженнях ці алгоритми дозволяють проводити експерименти в умовах, де присутність людини неможлива, наприклад, у космосі або під водою, відкриваючи нові можливості для відкриттів.

Таким чином, алгоритми візуалізації руху роботизованих платформ забезпечують підвищення точності, безпеки та ефективності в різних галузях. Їхній розвиток відкриває нові перспективи для автоматизації, адаптації до складних умов та підвищення автономності роботизованих систем. Це робить їх важливим інструментом сучасної робототехніки та автоматизації.

2 ПІДГОТОВКА ДО РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Програмні засоби для моделювання та тестування роботизованих платформ

Для виконання кваліфікаційної роботи було використано програмне забезпечення Robotino SIM Demo та Robotino View 2, які є ключовими інструментами для роботи з роботизованими платформами. Ці програмні продукти забезпечують широкий набір функцій для моделювання, управління, програмування та тестування роботів у різних умовах. Їхнє застосування дозволяє проводити експерименти, що максимально наближені до реальних ситуацій, і отримувати детальний аналіз ефективності алгоритмів, які були розроблені у рамках роботи.

Robotino SIM Demo є сучасним симуляційним середовищем, що дозволяє віртуально моделювати поведінку роботів у різних сценаріях. Це середовище забезпечує користувачів можливістю створювати інтерактивні моделі, тестувати траєкторії руху та оцінювати адаптивність алгоритмів до динамічних змін у середовищі. Завдяки цьому інструменту можна створювати різноманітні сценарії, які імітують реальні умови роботи, що робить його надзвичайно корисним для перевірки алгоритмів на предмет стабільності, точності та ефективності. Також Robotino SIM Demo підтримує інтеграцію із сенсорними даними, що дозволяє детально аналізувати, як платформа реагує на зовнішні подразники.

Robotino View 2 є важливим інструментом для роботи з реальними роботизованими платформами. Це програмне забезпечення дозволяє програмувати рух платформи, задавати параметри її роботи та забезпечувати інтеракцію із сенсорними системами. Завдяки широкому набору функцій, Robotino View 2 дає змогу не лише керувати платформою, але й здійснювати детальний моніторинг її поведінки в реальних умовах. Інструмент дозволяє

тестувати алгоритми в умовах, максимально наближених до практичного застосування, і надавати зворотній зв'язок щодо коригування параметрів для покращення результатів.

Ці інструменти забезпечують інтеграцію теоретичних знань із практичними завданнями, дозволяючи ефективно перевіряти працездатність та надійність розроблених алгоритмів. Наприклад, тестування алгоритмів в Robotino SIM Demo дає змогу виявити потенційні недоліки або слабкі місця ще до впровадження у реальні платформи. Надалі ці алгоритми можуть бути перенесені в середовище Robotino View 2, де вони проходять тестування вже на фізичних роботах. Це дозволяє отримати повну картину того, як алгоритми працюють в умовах реального світу.

Окрім цього, використання Robotino SIM Demo і Robotino View 2 сприяє підвищенню продуктивності роботи завдяки можливості паралельного моделювання та тестування. Це дає змогу значно зекономити час, адже віртуальна симуляція дозволяє швидко коригувати алгоритми, а вже оптимізовані рішення впроваджувати у реальне середовище. Завдяки інтеграції цих інструментів розробка алгоритмів руху для роботизованих платформ досягає високого рівня точності та надійності.

2.2 Детальна класифікація алгоритмів візуалізації руху

Існує кілька основних типів алгоритмів, які використовуються для візуалізації руху роботизованих платформ, кожен з яких володіє своїми унікальними перевагами і підходить для різних типів завдань. Ці алгоритми є критично важливими для забезпечення точного та безпечного руху роботів у складних середовищах. Вони варіюються за рівнем складності, використанням обчислювальних ресурсів та ступенем адаптивності до навколишнього середовища. Алгоритми можуть бути як простими і легкими для реалізації, так і складними, з урахуванням великої кількості зовнішніх факторів, таких як інерція, тертя або гравітація. Використання різних типів

алгоритмів дозволяє роботам ефективно виконувати завдання в різних умовах та галузях.

2.2.1 Кінематичні алгоритми

Ці алгоритми базуються на законах кінематики і використовують математичні моделі для розрахунку траєкторії руху. Вони застосовують рівняння, що враховують швидкість, прискорення і положення об'єкта в просторі. Це особливо корисно для роботів, які виконують завдання, що вимагають високої точності, наприклад, зварювання або монтаж мікросхем. Кінематичні алгоритми (рис. 2.7) легко реалізуються і не потребують великої кількості обчислювальних ресурсів, що робить їх ідеальним вибором для багатьох промислових застосувань [4], [16]. Незважаючи на їхню простоту, такі алгоритми здатні забезпечувати високий рівень точності у роботі.

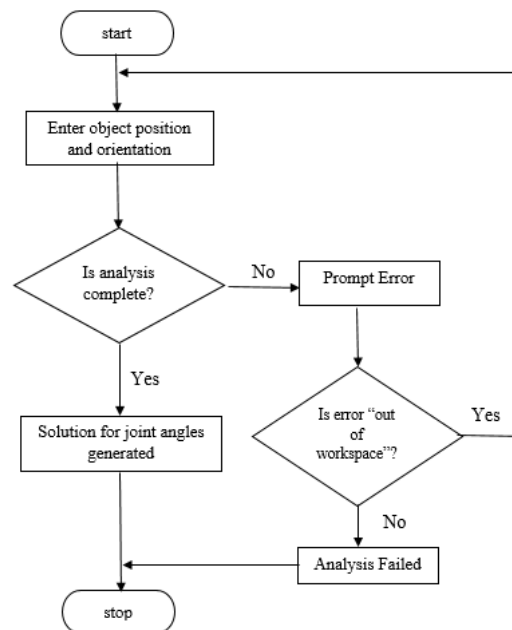


Рисунок 2.7 – Базовий кінематичний алгоритм [16]

Основні формули включають:

а) рівняння руху для визначення положення, швидкості і прискорення об'єкта:

$$v = v_0 + a \cdot t,$$

$$s = s_0 + v_0 \cdot t + 1/2 \cdot a \cdot t^2,$$

$$v^2 = v_0^2 + 2 \cdot a \cdot s,$$

де v – кінцева швидкість;

v_0 – початкова швидкість;

a – прискорення;

t – час;

s – переміщення;

б) позиція в просторі (x, y, z):

$$x(t) = x_0 + vx \cdot t,$$

$$y(t) = y_0 + vy \cdot t,$$

$$z(t) = z_0 + vz \cdot t,$$

де (x_0, y_0, z_0) – початкова позиція;

а (vx, vy, vz) – компоненти швидкості.

2.2.2 Динамічні алгоритми

Динамічні алгоритми (рис. 2.8) є складнішими за кінематичні, оскільки враховують не лише швидкість і прискорення, а й зовнішні сили, такі як інерція, тертя та гравітація. Вони забезпечують точніше управління роботизованою платформою, особливо в умовах, що постійно змінюються.

Ці алгоритми використовуються в середовищах, де взаємодія з об'єктами має ключове значення. Наприклад, у промисловості вони дозволяють враховувати вагу та розташування вантажів, а в автономному

транспорті – вплив дорожніх умов та розподілу навантаження. Завдяки цьому динамічні алгоритми забезпечують адаптацію роботів до складних зовнішніх факторів і підвищують їхню ефективність у реальних умовах.

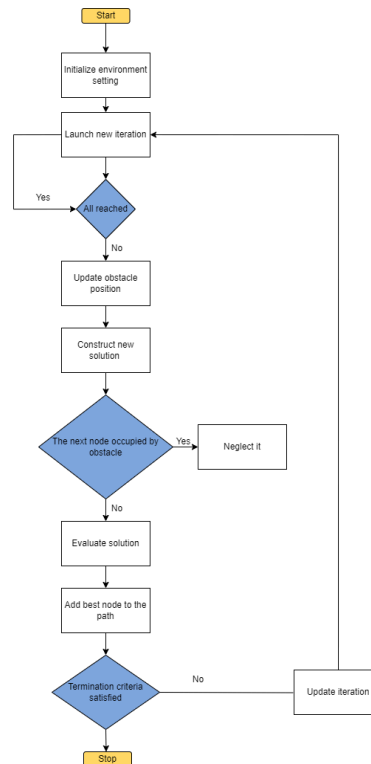


Рисунок 2.8 – Приклад динамічного алгоритму

Динамічні алгоритми враховують сили, що діють на об'єкт, і використовують закони динаміки Ньютона:

а) другий закон Ньютона:

$$F = m \cdot a,$$

де F – сила;

m – маса об'єкта;

a – прискорення;

б) обчислення сили тертя:

$$F_{\text{тертя}} = \mu \cdot N,$$

де μ – коефіцієнт тертя;

N – нормальна сила;

в) момент сили для обертання:

$$\tau = I \cdot \alpha,$$

де τ – момент сили;

I – момент інерції;

α – кутове прискорення.

2.2.3 Алгоритми на основі комп'ютерного зору

Алгоритми комп'ютерного зору (рис. 2.9) дозволяють відстежувати положення робота та об'єктів у його оточенні в режимі реального часу.

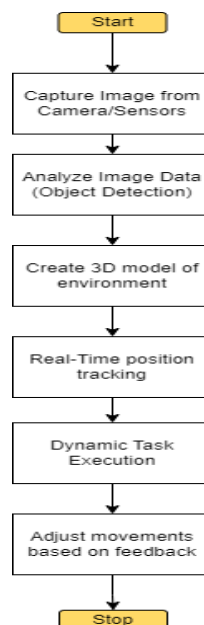


Рисунок 2.9 – Приклад алгоритму на основі комп'ютерного зору

Алгоритми комп'ютерного зору часто базуються на перетвореннях зображень, зокрема гомографіях та матрицях трансформацій:

а) матриця гомографії (перетворення координат між двома площинами):

$$p' = H \cdot p,$$

де p – точка на вихідному зображенні;

H – матриця гомографії;

p' – трансформована точка;

б) перетворення камери (для обчислення координат об'єкта):

$$World = K - 1 \cdot Camera,$$

де K – матриця камери (внутрішні параметри);

$Camera$ – координати у системі камери;

$World$ – координати у світовій системі.

2.2.4 Алгоритми на основі штучного інтелекту

Штучний інтелект (ШІ) відіграє важливу роль у вдосконаленні алгоритмів візуалізації руху.

Використання нейронних мереж і методів машинного навчання дозволяє роботизованим платформам адаптуватися до змін у навколишньому середовищі, самонавчатися і вдосконалювати свої навички під час роботи.

Алгоритми (рис. 2.10) на основі ШІ здатні прогнозувати траєкторію руху на основі накопичених даних і коригувати дії платформи у реальному часі. Завдяки цьому, роботи можуть швидко адаптуватися до нових умов, що робить їх незамінними у складних промислових і дослідницьких завданнях.

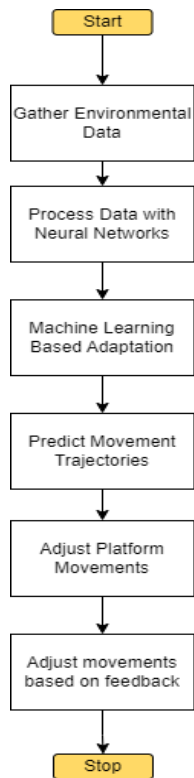


Рисунок 2.10 – Приклад алгоритму на основі III

Алгоритми III часто включають нейронні мережі та машинне навчання. Основні формули тут включають:

а) функція активації нейронної мережі:

$$a = f(w \cdot x + b),$$

де a – активація нейрона;

w – ваги;

x – вхідні дані;

b – зміщення (bias);

f – активаційна функція (наприклад, ReLU, Sigmoid);

б) функція помилки (loss function), яку нейронна мережа намагається мінімізувати:

$$L = 1/n \sum_{i=1}^n (y_i - \hat{y}_i)^2,$$

де L – середньоквадратична помилка;

y_i – справжнє значення;

$y_i^{\wedge}$ – прогнозоване значення;

n – кількість даних;

в) зворотнє поширення помилки (backpropagation) для оновлення ваг нейронної мережі:

$$w_{new} = w_{old} - \alpha \cdot \nabla L(w),$$

де α – швидкість навчання;

$\nabla L(w)$ – градієнт функції втрат за вагою w .

2.2.5 Алгоритми віртуальної та доповненої реальностей

Використання технологій віртуальної реальності (VR) і доповненої реальності (AR) (рис. 2.11) дозволяє операторам взаємодіяти з віртуальними моделями роботів у режимі реального часу.

Ці алгоритми допомагають точніше управляти роботами, оскільки оператор може бачити траєкторію руху у контексті реального світу.

VR дозволяє моделювати рух роботів в умовах, наближених до реальних, що є надзвичайно корисним для навчання і тестування роботів перед їх використанням у реальних умовах.

AR, у свою чергу, накладає віртуальні об'єкти на фізичний світ, дозволяючи операторам візуалізувати траєкторії руху і приймати рішення на основі реальних даних.

Таким чином, різноманітні алгоритми візуалізації руху роботизованих платформ дозволяють забезпечувати високий рівень точності, адаптивності та ефективності при виконанні різноманітних завдань у складних умовах.

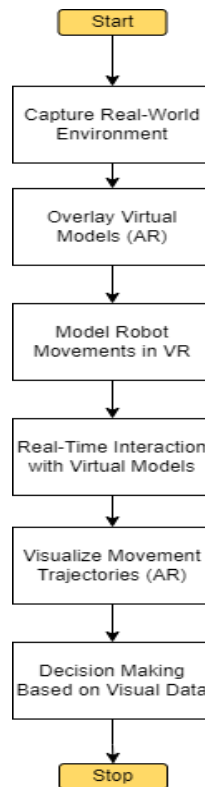


Рисунок 2.11 – Приклад алгоритму на основі віртуальної та доповненої реальностей

Алгоритми віртуальної та доповненої реальностей використовують геометричні перетворення та методи відтворення віртуальних сцен.

а) матриця масштабування: Для цього може використовуватися матриця масштабування з коефіцієнтами масштабування по осях. Або поворот у 3D просторі також відомий як кут Ейлера. І також може використовуватися перспективна проекція для відображення 3D-сцен на 2D-екрані

2.2.6 Алгоритми навігації

Алгоритми навігації (рис. 2.12) відповідають за здатність роботизованих платформ орієнтуватися в просторі, планувати свій шлях та уникати перешкод під час руху. Ці алгоритми об'єднують інформацію з сенсорів (лідарів, камер, радарів) для створення карти середовища та розрахунку оптимального маршруту. Основні завдання навігаційних алгоритмів включають:

- SLAM (Simultaneous Localization and Mapping) – одночасне локалізування робота в середовищі та побудова карти;
- алгоритми уникнення перешкод – використовують сенсори для виявлення об'єктів на шляху робота і корекції траєкторії;
- алгоритми планування маршруту – розраховують найкоротший або найефективніший маршрут від початкової до кінцевої точки. Прикладами є алгоритми Dijkstra, A*, або Rapidly-exploring Random Tree (RRT).

Ці алгоритми широко використовуються в автономних транспортних засобах, логістиці (наприклад, роботах на складах) та виробничих роботах.

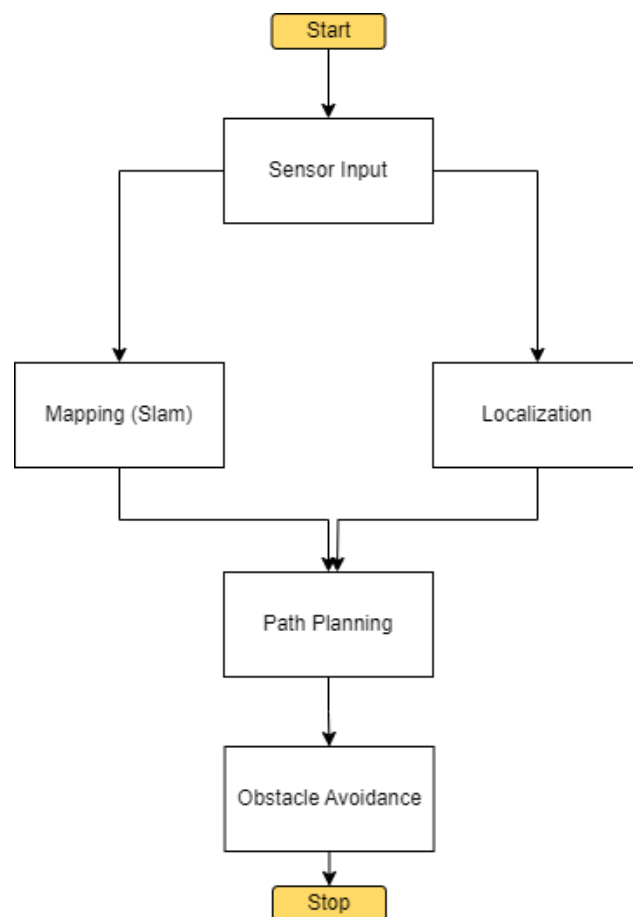


Рисунок 2.12 – Приклад алгоритму навігації

Алгоритми навігації включають планування траєкторії та уникнення перешкод. Основні формули включають:

- а) алгоритм Дейкстри для пошуку найкоротшого шляху.

Вагова функція для пошуку найкоротшого шляху:

$$d(u, v) = \min(d(u, v) + w(u, v)),$$

де (u, v) – відстань між вузлами u і v ;

$w(u, v)$ – вага ребра між вузлами;

б) алгоритм A^* (A-star) для пошуку шляху з евристикою.

Функція оцінки:

$$f(n) = g(n) + h(n),$$

де $f(n)$ – оцінка вартості поточного шляху;

$g(n)$ – реальна вартість від початку до вузла n ;

$h(n)$ – евристична оцінка вартості від вузла n до цілі;

в) уникнення перешкод.

Потенційні поля:

$$F_{repulsive} = k \cdot (1/d^2),$$

де $F_{repulsive}$ – виштовхувальна сила;

d – відстань до перешкоди;

k – константа відштовхування.

2.2.7 Алгоритми зворотної кінематики

Алгоритми зворотної кінематики (рис. 2.13) дозволяють розрахувати, як повинні рухатися всі частини роботизованої платформи (або маніпулятора) для досягнення заданого положення в просторі. Це включає розрахунок необхідних кутів суглобів робота на основі кінцевого положення руки або інструменту. Такі алгоритми важливі для:

- роботизованих маніпуляторів, які виконують точні операції, такі як зварювання або складання;
- медичних роботів під час хірургічних операцій, де точність є критичною.

Зворотна кінематика використовується також у анімації роботів та тренажерах для моделювання реальних умов роботи. Ці алгоритми дозволяють забезпечити плавний і точний рух маніпуляторів навіть у складних середовищах.

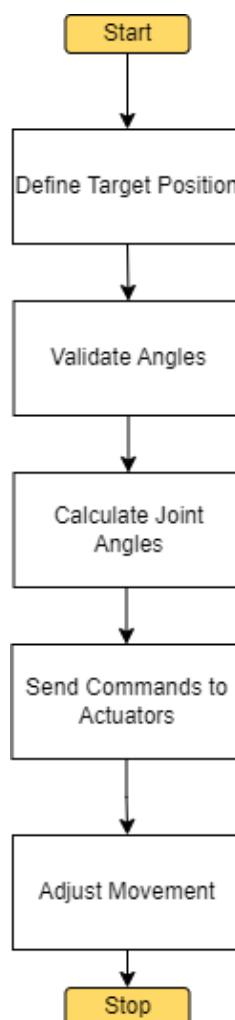


Рисунок 2.13 – Приклад алгоритму зворотної кінематики

а) зворотне перетворення матриць для маніпуляторів:

$$\theta = \text{atan2}(y, x),$$

де θ – кут суглоба;

x, y – координати кінцевої точки;

б) загальні рівняння зворотної кінематики:

$$\theta_1 = \text{atan2}(y, x),$$

$$\theta_2 = \arccos(x^2 + y^2 - l_1^2 - l_2^2 / 2 \cdot l_1 \cdot l_2),$$

де l_1, l_2 – довжини сегментів маніпулятора.

2.2.8 Алгоритми сенсорного злиття (Sensor Fusion Algorithms)

Алгоритми сенсорного злиття (рис. 2.14) об'єднують дані з різних сенсорів, таких як камери, лідари, радары, GPS та інерційні датчики, для створення більш точного уявлення про навколишнє середовище та прийняття ефективних рішень.

Кожен сенсор має свої переваги та обмеження, тому поєднання їхніх даних дозволяє:

- компенсувати слабкі сторони одного сенсора за рахунок сильних сторін іншого;
- підвищити точність локалізації та навігації робота в просторі;
- поліпшити якість виявлення об'єктів, оптимізувати ухвалення рішень та адаптацію до змін середовища.

Такі алгоритми широко застосовуються в автономних транспортних засобах, які потребують надійної навігації та безпеки, у роботах для дослідження складних середовищ (наприклад, підводних або космічних місій), а також у виробничих системах, де необхідно точно відстежувати рух та взаємодію з об'єктами в реальному часі.

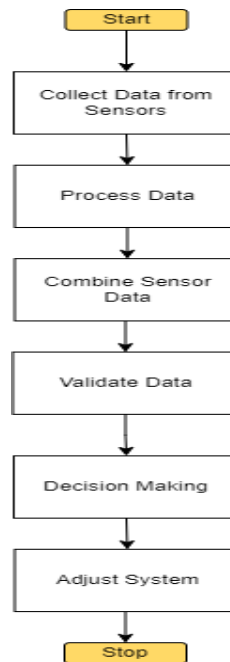


Рисунок 2.14 – Приклад алгоритму сенсорного злиття

Для сенсорного злиття використовуються методи для об'єднання даних із кількох джерел, наприклад, фільтр Калмана:

а) фільтр Калмана.

Оновлення стану:

$$x_k = A \cdot x_{k-1} + B \cdot u_k + w_k;$$

Оновлення спостереження:

$$z_k = H \cdot x_k + v_k,$$

де x_k – стан системи;

A – матриця переходу;

B – матриця впливу керуючої дії;

u_k – керуюча дія;

w_k, v_k – шум процесу і вимірювання.

2.2.9 Алгоритми прогнозування поведінки (Behavior Prediction Algorithms)

Алгоритми прогнозування поведінки (рис. 2.15) допомагають автономним транспортним засобам або роботам передбачати майбутні дії інших об'єктів у середовищі, таких як пішоходи, інші автомобілі чи роботи. Ці алгоритми аналізують:

- швидкість і напрямок руху об'єктів;
- моделі поведінки людей або інших транспортних засобів на основі попередніх даних;
- контекст середовища, наприклад, перехрестя, світлофори або пішохідні переходи.

Прогнозування поведінки дозволяє автономним системам запобігати аваріям, швидко ухвалювати рішення і рухатися в складних умовах.

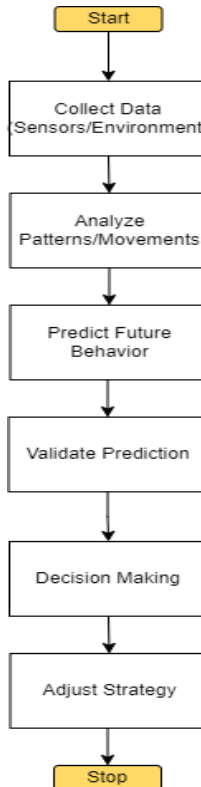


Рисунок 2.15 – Приклад алгоритму прогнозування поведінки

Прогнозування поведінки може базуватися на статистичних моделях або моделюванні траєкторій:

а) прогнозування траєкторії:

$$x^{\wedge}(t) = x_0 + v \cdot t + 1/2 \cdot a \cdot t^2,$$

де $x^{\wedge}(t)$ – прогнозоване положення в момент часу t ;

x_0 – початкове положення;

v – швидкість;

a – прискорення;

б) Баєсівське прогнозування:

$$P(x|y) = P(y|x) \cdot P(x) / P(y),$$

де $P(x|y)$ – ймовірність прогнозованої поведінки;

$P(y|x)$ – ймовірність даних, якщо поведінка відома.

2.2.10 Алгоритми оптимізації руху (Motion Optimization Algorithms)

Алгоритми оптимізації руху (рис. 2.16) відповідають за максимальну ефективність руху роботизованих платформ або транспортних засобів. Вони розраховують оптимальні траєкторії та режими руху, враховуючи такі фактори, як:

- мінімізація енергоспоживання – вибір траєкторії або швидкості, яка знижує витрати енергії;
- швидкість виконання завдань – вибір найшвидшого або найефективнішого способу виконання роботи;
- безпека руху – вибір траєкторії з урахуванням уникнення ризиків або перешкод.

Ці алгоритми застосовуються в роботах на виробничих лініях, автономних транспортних засобах та інших системах, де ефективність та точність виконання завдань критично важливі.

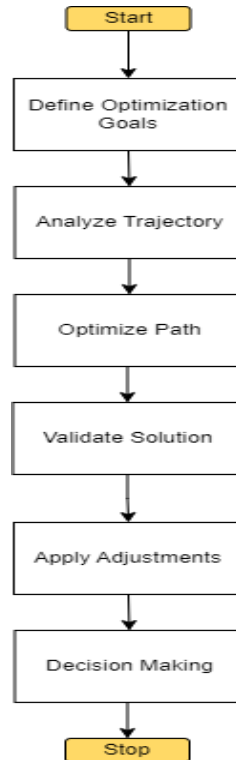


Рисунок 2.16 – Приклад алгоритму оптимізації руху

Для оптимізації руху використовуються методи мінімізації функції витрат або шляху:

Функція витрат:

$$J = \int_0^T (1/2 \cdot \|u(t)\|^2) dt,$$

де J – функція витрат;

$u(t)$ – керуючий сигнал;

T – тривалість оптимізації.

Гradientний спуск для мінімізації функції:

$$x_{k+1} = x_k - \alpha \cdot \nabla f(x_k).$$

де x_{k+1} – поточне положення;

α – швидкість навчання;

$\nabla f(x_k)$ – градієнт функції витрат.

2.3 Проблеми та виклики в реалізації алгоритмів візуалізації

Незважаючи на значні переваги алгоритмів візуалізації, їх впровадження супроводжується певними проблемами та викликами, які необхідно враховувати під час розробки і використання таких систем. Ці виклики пов'язані з обробкою великих обсягів даних, забезпеченням високої точності та стабільності, адаптацією до умов навколишнього середовища, інтеграцією з іншими системами, а також ефективністю енергоспоживання. Кожен з цих аспектів потребує ретельного планування, оптимізації та постійного вдосконалення, щоб забезпечити надійність і функціональність роботизованих платформ у складних умовах. Успішне подолання цих викликів дозволяє значно підвищити продуктивність роботизованих систем і розширити можливості їх застосування в різних галузях.

2.3.1 Обробка великих обсягів даних

Одна з головних проблем, з якою стикаються алгоритми візуалізації руху, полягає в необхідності обробки величезних масивів інформації, що надходить від численних сенсорів, камер та інших пристроїв. Це створює значне навантаження на обчислювальні системи, що може призвести до затримок у прийнятті рішень, особливо в режимі реального часу. Високі вимоги до ресурсів збільшують вартість систем та можуть обмежити їх

ефективність у випадках, коли доступ до обчислювальних потужностей є обмеженим. Для забезпечення швидкої та надійної обробки даних необхідно розробляти оптимізовані алгоритми, здатні працювати ефективно навіть у випадках великого навантаження.

2.3.2 Точність та стабільність

Для виконання високоточних операцій, таких як хірургічні втручання або збирання мікроскопічних компонентів, необхідна максимальна точність візуалізаційних алгоритмів. Однак навіть найменші відхилення у розрахунках або невідповідності в роботі сенсорів можуть призвести до критичних помилок. Це вимагає постійного калібрування та вдосконалення алгоритмів для забезпечення стабільної та надійної роботи систем. Високі вимоги до точності створюють додаткові виклики для розробників, оскільки помилки в цих випадках можуть мати серйозні наслідки, особливо в критичних галузях, таких як медицина.

2.3.3 Залежність від навколишнього середовища

Алгоритми візуалізації часто залежать від зовнішніх умов, що може створювати додаткові проблеми. Наприклад, зміни освітлення, наявність пилу, диму або інших факторів можуть впливати на роботу сенсорів та камер, що ускладнює точність візуалізації. У таких випадках необхідно використовувати адаптивні алгоритми, які можуть змінювати свої налаштування в залежності від умов навколишнього середовища. Це забезпечує стабільну роботу роботизованих платформ навіть у складних або непередбачуваних умовах, що є важливим для їх успішного використання у промислових та дослідницьких завданнях.

2.3.4 Інтеграція з іншими системами

Ефективна робота алгоритмів візуалізації вимагає їх інтеграції з іншими системами, такими як системи управління та безпеки. Це може створювати

труднощі, оскільки різні типи алгоритмів можуть конфліктувати між собою або вимагати складної координації для забезпечення злагодженої роботи. Успішна інтеграція дозволяє забезпечити безперебійну роботу всієї системи, що є особливо важливим у складних роботизованих платформах, де помилки у взаємодії між компонентами можуть призвести до серйозних збоїв.

2.3.5 Енергоефективність

Автономні роботизовані системи зазвичай працюють на обмежених енергетичних ресурсах, що створює виклики щодо енергоефективності алгоритмів візуалізації. Процес обробки даних може вимагати значних енергетичних витрат, що обмежує тривалість роботи роботів без підзарядки. Для вирішення цієї проблеми необхідно розробляти алгоритми, які не тільки забезпечують високу точність, але й оптимізують використання енергії. Це дозволить значно збільшити тривалість автономної роботи систем і підвищити їх ефективність у довготривалих завданнях.

Таким чином, попри значні переваги, алгоритми візуалізації стикаються з низкою викликів, пов'язаних з обробкою даних, точністю, впливом середовища, інтеграцією та енергоефективністю. Однак ці проблеми можуть бути вирішені за допомогою вдосконалення алгоритмів, оптимізації обчислювальних процесів та розробки адаптивних систем, здатних працювати в складних умовах.

2.4 Порівняльний аналіз ефективності різних алгоритмів в умовах динамічного середовища

В цьому пункті буде проведено аналіз ефективності різних алгоритмів в умовах динамічного середовища для того щоб визначити які алгоритми найбільш ефективні в тих чи інших умовах.

2.4.1. Кінематичні алгоритми

Швидкість обробки – висока. Кінематичні алгоритми базуються на простих рівняннях руху (швидкість, прискорення) і не потребують обробки складних даних або взаємодії з сенсорами.

Точність руху – обмежена. Ці алгоритми добре працюють у стабільних умовах, де не враховуються зовнішні сили (інерція, тертя), що робить їх менш ефективними в умовах динамічного середовища.

Адаптивність – низька. Кінематичні алгоритми не адаптуються до змін навколишнього середовища, оскільки не враховують динамічних змін, таких як взаємодія з іншими об'єктами.

Складність реалізації – низька. Кінематичні алгоритми прості у реалізації та потребують мінімальних обчислювальних ресурсів.

2.4.2. Динамічні алгоритми

Швидкість обробки – помірна. Динамічні алгоритми враховують сили, що діють на роботизовані платформи, що вимагає більших обчислювальних ресурсів.

Точність руху – висока. Враховують сили інерції, тертя та інші зовнішні сили, що дозволяє точно контролювати рух у динамічному середовищі.

Адаптивність – висока. Ці алгоритми можуть швидко адаптуватися до змін у навколишньому середовищі та забезпечують точне управління навіть при зміні умов.

Складність реалізації – висока. Вимагають значних обчислювальних ресурсів для обчислення сили, прискорення та інших параметрів руху.

2.4.3. Алгоритми комп'ютерного зору

Швидкість обробки – залежить від апаратного забезпечення. Обробка зображень займає значний час, особливо для високої роздільної здатності або обробки великих даних з багатьох сенсорів.

Точність руху – висока. Алгоритми комп'ютерного зору забезпечують точне розпізнавання об'єктів та орієнтацію в просторі, що робить їх ефективними для навігації у складних умовах.

Адаптивність – висока. Можуть адаптуватися до динамічних змін у навколишньому середовищі (нові об'єкти, зміна освітлення тощо).

Складність реалізації – висока. Потребує значних ресурсів для обробки відеопотоків і розпізнавання об'єктів у реальному часі.

2.4.4. Алгоритми на основі штучного інтелекту (ШІ)

Швидкість обробки – помірна. Швидкість залежить від архітектури нейронної мережі та кількості шарів, однак навчання може бути дуже тривалим.

Точність руху – висока. ШІ здатний вчитися з досвіду і покращувати свої рішення з часом, що забезпечує високу точність дій у динамічному середовищі.

Адаптивність – дуже висока. Алгоритми ШІ можуть адаптуватися до нових ситуацій та змін у середовищі, зокрема, до нових перешкод чи об'єктів;

Складність реалізації – висока. Потрібні значні обчислювальні ресурси для тренування моделей та виконання прогнозів у реальному часі.

2.4.5. Алгоритми сенсорного злиття

Швидкість обробки – висока. Оптимізовані для злиття даних з декількох джерел у реальному часі.

Точність руху – висока. Об'єднання даних із кількох сенсорів дає точніше уявлення про навколишнє середовище, покращуючи точність руху та орієнтації.

Адаптивність – висока. Можуть ефективно реагувати на динамічні зміни в середовищі завдяки інтеграції даних від різних сенсорів.

Складність реалізації – середня. Потребує наявності різних сенсорів і складної логіки для поєднання та інтерпретації даних.

2.4.6. Алгоритми прогнозування поведінки

Швидкість обробки – помірна. Швидкість залежить від складності моделі прогнозування (лінійні моделі працюють швидше, ніж глибокі нейронні мережі).

Точність руху – висока. Прогнозування майбутніх дій або руху об'єктів дає можливість завчасно ухвалювати рішення та коригувати рух.

Адаптивність – висока. Моделі прогнозування можуть адаптуватися до нових патернів поведінки, що робить їх корисними в динамічних середовищах.

Складність реалізації – висока. Необхідна наявність великих обсягів даних для побудови моделей поведінки, а також складних алгоритмів для прогнозування.

2.4.7. Алгоритми оптимізації руху

Швидкість обробки – залежить від методу оптимізації. Наприклад, методи градієнтного спуску можуть працювати швидко, тоді як генетичні алгоритми займають більше часу.

Точність руху – висока. Оптимізовані траєкторії забезпечують мінімізацію енерговитрат або часу на виконання завдань.

Адаптивність – помірна. Алгоритми можуть бути менш ефективними при різкій зміні умов, оскільки оптимізація відбувається для конкретної ситуації.

Складність реалізації – висока. Потребує значних обчислювальних ресурсів для пошуку оптимального рішення.

2.4.8. Алгоритми віртуальної та доповненої реальностей (VR/AR)

Швидкість обробки – залежить від апаратного забезпечення. Для відображення складних сцен в режимі реального часу потрібні потужні обчислювальні ресурси.

Точність руху – висока. Доповнена реальність забезпечує точне накладення віртуальних об'єктів на фізичний світ, що дозволяє точно керувати рухом.

Адаптивність – висока. Системи AR/VR можуть легко адаптуватися до змін у навколишньому середовищі завдяки постійній обробці даних з камер і сенсорів руху.

Складність реалізації – висока. Необхідна складна логіка для синхронізації віртуальних об'єктів із реальними, а також значна обчислювальна потужність для обробки відео та трекінгу в реальному часі.

2.4.9. Алгоритми навігації

Швидкість обробки – залежить від алгоритму. Алгоритми типу Dijkstra можуть бути повільними в складних графах, тоді як алгоритм A* з евристиками працює швидше.

Точність руху – висока. Алгоритми навігації забезпечують точний рух по запланованій траєкторії, зокрема для уникнення перешкод.

Адаптивність – висока. Алгоритми навігації з можливістю динамічної зміни маршруту здатні швидко адаптуватися до нових умов (перешкод, зміни середовища).

Складність реалізації – середня/висока. Залежить від складності середовища і вибраного алгоритму (Dijkstra, A*, SLAM). Для динамічних середовищ і автономних систем навігація потребує складної реалізації з багатьма сенсорами.

2.4.10. Алгоритми зворотної кінематики

Швидкість обробки – висока для двовимірних систем і простих маніпуляторів. Для багатовимірних систем обчислення складніші і можуть вимагати більше часу.

Точність руху – висока. Алгоритми зворотної кінематики точно обчислюють необхідні кути суглобів, щоб досягти заданого положення в просторі.

Адаптивність – помірна. Алгоритми зворотної кінематики можуть швидко пристосовуватися до зміни положення кінцевої точки, але менш ефективні при врахуванні динамічних зовнішніх умов (сили, тертя).

Складність реалізації – висока для багатозв'язкових систем, оскільки для кожного суглоба потрібні точні обчислення і оптимізація.

2.5 Порівняльна таблиця

Таблиця детально аналізує та порівнює ефективність різних алгоритмів візуалізації руху роботизованих платформ у динамічному середовищі за кількома ключовими показниками.

Вона є важливим інструментом для прийняття рішень, який дозволяє визначити найоптимальніший підхід залежно від специфічних вимог, завдань та умов експлуатації платформи.

Такий підхід допомагає врахувати не лише продуктивність алгоритму, але й його сумісність із технічними та програмними ресурсами. Основними критеріями порівняння виступають:

- Швидкість обробки – показник того, наскільки оперативно алгоритм може обробляти вхідні дані та реагувати на зміни в режимі реального часу.

Висока швидкість обробки є ключовою вимогою для ефективного функціонування в динамічному середовищі, де події змінюються швидко та непередбачувано. Затримки в обробці можуть суттєво вплинути на точність виконання завдань або навіть зробити систему непридатною до роботи у певних умовах.

- Точність руху – здатність алгоритму забезпечувати точне переміщення роботизованої платформи навіть у складному середовищі.

Це включає виконання маневрів із високою точністю, уникнення перешкод, дотримання заданих траєкторій руху та мінімізацію відхилень від цільових параметрів. Від точності алгоритму залежить успішність виконання завдань, особливо у випадках, де помилки недопустимі.

– Адаптивність – характеристика, яка визначає, наскільки ефективно алгоритм може підлаштовуватися до змін у навколишньому середовищі. Це включає здатність реагувати на появу нових перешкод, адаптуватися до змін освітлення, умов рельєфу чи інших факторів, які можуть виникати під час виконання завдань.

Адаптивність особливо важлива для роботів, які працюють у реальному часі, оскільки це значно підвищує їхню автономність та функціональність.

– Складність реалізації – оцінка технічних вимог до впровадження алгоритму.

Це включає потребу в обчислювальних ресурсах, сумісність із наявним апаратним забезпеченням, а також складність програмної реалізації. Деякі алгоритми, хоч і демонструють високу продуктивність, можуть вимагати значних витрат на їх розробку та інтеграцію.

Таблиця (табл. 2.1) не лише надає загальне уявлення про переваги та недоліки кожного алгоритму, але й дозволяє прийняти обґрунтоване рішення щодо їх вибору для конкретних завдань.

Наприклад, у задачах, де критичною є швидка реакція системи, перевага надається алгоритмам із високою швидкістю обробки, навіть якщо вони мають деякі компроміси в точності.

Водночас, у середовищах із високими вимогами до точності та адаптивності, можуть використовуватися алгоритми, які вимагають більше обчислювальних ресурсів та мають більшу складність у реалізації.

Загалом, кожен алгоритм має свої сильні та слабкі сторони, які обумовлюють його ефективність у різних умовах експлуатації.

Ретельний аналіз цих характеристик допомагає обрати найбільш відповідний алгоритм для досягнення поставлених цілей, забезпечуючи оптимальну продуктивність і надійність роботизованої платформи.

Таблиця 2.1 – Результати аналізу алгоритмів візуалізації

Алгоритм	Швидкість обробки	Точність руху	Адаптивність	Складність реалізації
Кінематичні алгоритми	Висока	Обмежена	Низька	Низька
Динамічні алгоритми	Помірна	Висока	Висока	Висока
Алгоритми комп'ютерного зору	Залежить	Висока	Висока	Висока
Алгоритми на основі ІІІ	Помірна	Висока	Дуже висока	Висока
Алгоритми сенсорного злиття	Висока	Висока	Висока	Середня
Алгоритми прогнозування поведінки	Помірна	Висока	Висока	Висока
Алгоритми оптимізації руху	Залежить	Висока	Помірна	Висока
Алгоритми віртуальної та доповненої реальностей	Залежить	Висока	Висока	Висока
Алгоритми навігації	Залежить	Висока	Висока	Середня/Висока
Алгоритми зворотної кінематики	Висока для простих випадків	Висока	Помірна	Висока

Висновок з таблиці:

- Алгоритми віртуальної та доповненої реальностей добре працюють у динамічних середовищах, дозволяючи створювати інтерактивні моделі, але вони вимагають високої обчислювальної потужності для обробки даних у реальному часі, що може обмежувати їх використання на слабких пристроях;
- Алгоритми навігації особливо корисні для автономних систем, забезпечуючи високу точність та адаптивність, зокрема в умовах змін середовища, таких як уникнення перешкод, або роботи у складних та непередбачуваних умовах;

– Алгоритми зворотної кінематики забезпечують високу точність у керуванні маніпуляторами, що робить їх ідеальними для точних операцій, але вони можуть бути менш ефективними при раптових зовнішніх змінах, таких як змінне навантаження чи рухома ціль.

2.6 Технологічні перспективи та можливості розвитку

Алгоритми візуалізації руху роботизованих платформ продовжують стрімко розвиватися, відкриваючи нові перспективи для їх використання в різних галузях. Ключові напрямки розвитку цих алгоритмів спрямовані на підвищення точності, автономності та ефективності роботизованих систем. Завдяки інтеграції з сучасними технологіями, такими як штучний інтелект, сенсорика, віртуальна та доповнена реальність, можливості візуалізації руху значно розширюються. Це дозволяє створювати системи, здатні виконувати складні операції з мінімальним втручанням людини, підвищуючи продуктивність і безпеку в промисловості, медицині, логістиці та інших сферах. Розвиток цих технологій відкриває нові горизонти для впровадження роботизованих систем на великих підприємствах та у складних інфраструктурних проектах, роблячи їх незамінними в сучасному технологічному світі.

2.6.1 Інтеграція зі штучним інтелектом

Одним із ключових напрямків розвитку алгоритмів візуалізації є їх інтеграція з технологіями штучного інтелекту (ШІ).

Завдяки машинному навчанню роботизовані платформи (рис. 2.17) можуть самонавчатися під час роботи, аналізувати поведінку у реальному часі та коригувати траєкторії руху в залежності від змін у середовищі [4], [17].

Це дозволяє не лише підвищити точність і швидкість роботи, але й ефективно передбачати можливі проблеми, що виникають у процесі виконання завдань.

ІІІ також дозволяє роботам швидше реагувати на зміни в оточенні, що підвищує рівень автономності роботизованих платформ і зменшує потребу у втручанні людини.



Рисунок 2.17 – Автоматизовані роботи зі штучним інтелектом в промисловості [17]

2.6.2 Підвищення точності сенсорів

Інновації у сфері сенсорики, зокрема розвиток технологій LIDAR і глибоких камер, забезпечують отримання все більш точних і детальних тривимірних моделей простору. Ці вдосконалені сенсори дозволяють роботам краще орієнтуватися в складних середовищах, підвищуючи точність рухів і зменшуючи кількість помилок. Вони працюють у поєднанні з потужними алгоритмами обробки даних, що робить можливим досягнення високої стабільності та надійності у роботі систем. Це є важливим фактором для впровадження роботизованих платформ у сфери, де необхідна висока точність, такі як медицина або промислові процеси.

2.6.3 Використання віртуальної та доповненої реальності

Технології віртуальної (VR) і доповненої реальності (AR) відкривають нові можливості для керування роботизованими платформами. За допомогою VR оператори можуть управляти роботами на відстані, моделюючи їхні дії у віртуальних середовищах, що імітують реальні умови. Це дозволяє операторам тренуватися без ризику пошкодження обладнання або збою в роботі систем. У свою чергу, AR дозволяє візуалізувати траєкторії руху роботів у реальному середовищі, накладаючи віртуальні об'єкти на фізичний світ. Це значно спрощує управління роботами та покращує навчання операторів, забезпечуючи більш інтуїтивну взаємодію з системами.

2.6.4 Підвищення рівня автономності

Один із найважливіших напрямків розвитку – це збільшення рівня автономності роботизованих платформ. Алгоритми візуалізації все частіше використовуються для розробки систем, здатних працювати без постійного втручання людини. Це відкриває нові можливості для використання роботів у віддалених або небезпечних середовищах, таких як дослідження океанів або космосу, де людська присутність є небажаною або неможливою. Автономні системи дозволяють роботам приймати рішення в режимі реального часу, адаптуючись до змін у навколишньому середовищі, що значно розширює спектр їхнього застосування.

2.6.5 Масштабованість

Розвиток алгоритмів візуалізації також сприяє підвищенню масштабованості роботизованих платформ. Це дозволяє використовувати роботів на великих виробництвах та в складних інфраструктурних проектах. Можливість управляти великими флотами роботів, які працюють разом для досягнення спільних цілей, робить такі системи незамінними у сучасній індустрії. Масштабованість роботизованих платформ дозволяє значно

підвищити продуктивність і забезпечує більш гнучке управління процесами, що є ключовим фактором у великих виробничих або логістичних проектах.

Таким чином, розвиток алгоритмів візуалізації руху роботизованих платформ відкриває нові можливості для автоматизації та підвищення ефективності різних процесів. Інтеграція зі штучним інтелектом, розвиток сенсорних технологій, використання VR і AR, підвищення рівня автономності та масштабованість роботизованих платформ — усе це сприяє розширенню сфер застосування роботів та їхньому впровадженню у найрізноманітніші галузі.

2.7 Проблеми та виклики в реалізації алгоритмів візуалізації

Обмежена обчислювальна потужність:

- виклик: реалізація складних алгоритмів, таких як штучний інтелект або динамічні алгоритми, вимагає значних обчислювальних ресурсів, особливо в реальному часі;

- проблема: недостатня потужність процесорів або апаратне обмеження може призвести до затримок у роботі, що є критичним для автономних систем;

- можливе рішення: використання прискорювачів (GPU, TPU) або оптимізованих алгоритмів для швидкої обробки.

Висока залежність від точності сенсорів:

- виклик: більшість алгоритмів візуалізації руху залежить від точності та якості сенсорних даних (лідарів, камер, IMU);

- проблема: неточність сенсорів, шум або погані умови роботи (наприклад, низьке освітлення, погана видимість) можуть впливати на точність алгоритмів;

- можливе рішення: використання алгоритмів сенсорного злиття для поєднання даних з декількох джерел і зниження впливу шуму.

Реалізація в реальному часі:

- виклик: багато алгоритмів (особливо на основі комп'ютерного зору та штучного інтелекту) повинні виконуватися в режимі реального часу, щоб забезпечити оперативну реакцію роботизованих платформ;

- проблема: алгоритми з високими вимогами до обчислювальних ресурсів можуть не впоратися з обробкою даних у реальному часі;

- можливе рішення: оптимізація кодів або застосування методів прогнозування для компенсації затримок в обробці.

Адаптивність до динамічного середовища:

- виклик: динамічні зміни в середовищі, такі як поява нових об'єктів або перешкод, можуть ускладнювати візуалізацію і планування руху;

- проблема: багато алгоритмів (особливо кінематичні) не здатні швидко адаптуватися до змін середовища;

- можливе рішення: використання алгоритмів штучного інтелекту та прогнозування поведінки для швидкої адаптації до нових умов.

Невизначеність та шум у даних:

- виклик: сенсори генерують шумні дані або дані з невизначеністю (особливо у складних середовищах, таких як пил, вода або погане освітлення);

- проблема: невизначеність може негативно впливати на точність моделі та спричиняти неправильні рішення в роботі алгоритмів;

- можливе рішення: застосування методів фільтрації (фільтр Калмана, частковий фільтр) або підвищення надійності даних через сенсорне злиття.

Робота в умовах складного рельєфу:

- виклик: для алгоритмів навігації та управління роботами складний рельєф може створювати додаткові проблеми через відсутність чітких орієнтирів або несподівані перешкоди;

- проблема: стандартні алгоритми навігації можуть не впоратися з умовами складного рельєфу, наприклад, на пересіченій місцевості;

- можливе рішення: використання картування в реальному часі (SLAM), алгоритмів глибинного навчання для передбачення перешкод і адаптації руху.

Висока складність алгоритмів штучного інтелекту:

- виклик: алгоритми на основі ШІ (нейронні мережі, глибоке навчання) часто складні у реалізації та вимагають величезних обсягів даних для навчання;

- проблема: процес навчання ШІ займає багато часу, і виникають труднощі з його застосуванням для реальних сценаріїв;

- можливе рішення: використання попередньо навчених моделей або комбінування з класичними алгоритмами для прискорення навчання.

Інтеграція різних типів алгоритмів:

- виклик: інтеграція різних алгоритмів візуалізації, таких як алгоритми комп'ютерного зору, штучного інтелекту, сенсорного злиття та кінематики, може створювати труднощі у досягненні узгоджених результатів;

- проблема: різні алгоритми можуть мати конфліктні вимоги або затримки при об'єднанні результатів, що впливає на загальну продуктивність системи;

- можливе рішення: розробка ефективних стратегій синхронізації даних та використання адаптивних підходів для комбінування різних алгоритмів.

Планування руху в багатовимірних системах:

- виклик: алгоритми, такі як зворотна кінематика, можуть стати складними для маніпуляторів із великою кількістю ступенів свободи (багатозв'язкові роботи);

- проблема: висока складність розрахунків та потреба в оптимізації руху кожного суглоба можуть значно знижувати швидкість роботи;

- можливе рішення: використання оптимізованих алгоритмів зворотної кінематики (наприклад, ітеративні методи) для зниження обчислювальних витрат.

Проблеми безпеки та надійності:

- виклик: у разі помилок у алгоритмах візуалізації руху, автономні системи можуть зіткнутися з перешкодами або виконати неправильні дії, що може призвести до аварій або пошкоджень;

- проблема: високий ризик збоїв в умовах непередбачуваних ситуацій, особливо для автономних транспортних засобів і роботів, що працюють поруч з людьми;

- можливе рішення: реалізація системи резервування та засобів перевірки для підвищення надійності та безпеки.

Непередбачувані середовища:

- виклик: деякі алгоритми (наприклад, комп'ютерного зору) важко адаптувати до змінних погодних умов, таких як дощ, сніг або туман;

- проблема: у таких умовах видимість може бути обмежена, і це впливає на точність сенсорних даних;

- можливе рішення: використання різноманітних сенсорів (інфрачервоні або ультразвукові) для отримання надійних даних в умовах поганої видимості.

Висока вартість впровадження:

- виклик: реалізація складних алгоритмів візуалізації може бути дорогою через необхідність спеціального обладнання (наприклад, лідари, високопродуктивні комп'ютери);

- проблема: для багатьох малих проектів вартість реалізації алгоритмів на базі ШІ, високоточних сенсорів або обчислювальних платформ може бути непідйомною;

- можливе рішення: використання менш дорогих технологій, як-от камери замість лідарів, або хмарні обчислення для зниження витрат.

2.8 Висновки до розділу

У другому розділі було проаналізовано основні етапи підготовки до розробки програмного забезпечення для роботизованих платформ, а також визначено ключові алгоритми та програмні засоби, які сприяють підвищенню точності, адаптивності та ефективності систем візуалізації руху.

Використання програмного забезпечення Robotino SIM Demo і Robotino View 2 дозволило створити ефективну основу для тестування та моделювання алгоритмів у віртуальному середовищі та їх впровадження на реальних платформах.

Проведено огляд та класифікацію основних алгоритмів візуалізації руху:

- кінематичні та динамічні алгоритми для розрахунку траєкторій і взаємодії із зовнішніми силами;
- алгоритми комп'ютерного зору, сенсорного злиття та навігації, які забезпечують точну орієнтацію і адаптацію в динамічному середовищі;
- алгоритми штучного інтелекту, які підвищують автономність і дозволяють роботам навчатися з досвіду.

Визначено ключові виклики, які виникають під час реалізації алгоритмів, такі як обмежена обчислювальна потужність, залежність від точності сенсорів, складність інтеграції та висока вартість впровадження.

Проведено порівняльний аналіз алгоритмів, що дозволяє обрати оптимальний підхід залежно від умов роботи та поставлених завдань. Це сприяє обґрунтованому вибору алгоритмів для конкретних сценаріїв використання роботизованих платформ.

Загалом, проведений аналіз показав важливість комплексного підходу до розробки програмного забезпечення для роботизованих платформ, що включає використання сучасних технологій моделювання, тестування і адаптивних алгоритмів. Такий підхід забезпечує високу надійність і ефективність роботи платформ у складних умовах експлуатації.

3 ПОБУДОВА АЛГОРИТМІВ РОБОТИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Основні поняття та визначення

Розробка алгоритмів для роботизованих платформ – це складний і багатоступінчастий процес, який вимагає врахування багатьох чинників. Робота включає проєктування, моделювання, тестування та впровадження систем, які керують рухом платформи та забезпечують її адаптацію до змін середовища. Успішна реалізація залежить від точності алгоритмів, ефективного використання сенсорів і оптимізації роботи під обмежені апаратні ресурси.

Головними завданнями алгоритмів є: [18]

- а) забезпечення точного і стабільного руху платформи;
- б) адаптація до змінних умов (наприклад, перешкоди чи нерівності поверхні);
- в) мінімізація енергоспоживання та раціональне використання обчислювальних ресурсів.

У рамках цього проєкту застосовувалися Robotino SIM Demo і Robotino View 2 – програмні інструменти, що дозволяють моделювати і тестувати алгоритми як у симуляційних умовах, так і на реальних платформах. Їх використання надало можливість ефективно аналізувати продуктивність розроблених рішень.

3.2 Моделювання алгоритмів у Robotino SIM Demo

Для тестування алгоритмів використовувалося симуляційне середовище Robotino SIM Demo. Його головною перевагою є можливість моделювання складних сценаріїв без ризику для реального обладнання.

Перевірялися різні траєкторії та маневри, включаючи уникнення перешкод і проходження складних ділянок. Оцінювалася точність виконання завдань і стабільність роботи алгоритмів.

Проводилися тести з урахуванням шуму сенсорних даних, зміни рельєфу та інших змінних. Це допомогло визначити стійкість алгоритмів до впливу зовнішніх умов.

3.3 Інтеграція алгоритмів у Robotino View 2

Після успішного тестування в симуляції алгоритми були перенесені на реальну платформу за допомогою Robotino View 2. Це середовище дозволяє програмувати платформу, налаштовувати її параметри та контролювати роботу в реальних умовах.

Параметри, розроблені в симуляції, були адаптовані для реального обладнання. Наприклад, чутливість сенсорів та швидкість руху коригувалися відповідно до характеристик платформи.

Реалізовані алгоритми збирали та обробляли дані з камер, лідарів та інших сенсорів. Це забезпечило злагоджену роботу системи і дозволило платформі точно орієнтуватися в просторі.

Після завершення налаштування платформа була протестована в лабораторних умовах. Випробування включали навігацію між перешкодами, виконання маневрів і точне позиціонування в обмеженому просторі [19].

3.4 Висновки до розділу

У третьому розділі було розглянуто процес побудови алгоритмів роботи програмного забезпечення для роботизованих платформ, починаючи з базових понять і визначень та завершуючи їх впровадженням у реальних умовах.

Основні аспекти розробки алгоритмів:

- визначено головні завдання алгоритмів: забезпечення точності руху, адаптація до змінних умов та раціональне використання ресурсів;
- підкреслено важливість оптимізації роботи алгоритмів під обмежені апаратні ресурси, що є критичним для роботизованих систем.

Моделювання в середовищі Robotino SIM Demo:

- використання симуляційного середовища дозволило протестувати алгоритми в умовах, максимально наближених до реальних, без ризику пошкодження обладнання;
- проведено перевірку стійкості алгоритмів до впливу зовнішніх факторів, таких як шум сенсорних даних та зміни рельєфу.

Інтеграція алгоритмів у реальну платформу через Robotino View 2:

- успішно адаптовано алгоритми, протестовані в симуляції, до реального обладнання;
- здійснено інтеграцію сенсорів (камер, лідарів) для забезпечення точного орієнтування платформи у просторі;
- проведено лабораторні випробування, які підтвердили ефективність роботи алгоритмів у реальних умовах.

Загалом, реалізація алгоритмів у симуляційному середовищі та їх адаптація для реальних роботизованих платформ підтвердили ефективність обраного підходу. Розроблені рішення забезпечують високу точність, стабільність та адаптивність роботи роботизованої платформи, що відкриває можливості для подальшого впровадження та вдосконалення систем.

4 ОПИС РЕАЛІЗАЦІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Середовище розробки проекту

Для розробки програмного забезпечення управління роботизованою платформою були обрані Robotino SIM Demo та Robotino View 2 [20]. Ці інструменти забезпечили як моделювання алгоритмів у віртуальному середовищі, так і їх подальшу реалізацію на реальних роботах.

Robotino SIM Demo дозволяє створювати симуляції, що наближені до реальних умов. У ньому можна тестувати траєкторії руху, перевіряти, як робот реагує на перешкоди або зміну умов середовища, а також оптимізувати алгоритми. Цей інструмент дає змогу швидко побачити слабкі місця в логіці руху, що значно спрощує процес розробки.

Robotino View 2 використовується для роботи з реальними платформами. Завдяки цьому програмному забезпеченню можна налаштовувати параметри руху, інтегрувати дані сенсорів і перевіряти, як алгоритми працюють у практичних завданнях. Крім того, Robotino View 2 підтримує популярні мови програмування, як-от Python і C++, що робить процес розробки більш гнучким і зручним.

4.2 Реалізація алгоритмів

Реалізація алгоритмів для управління платформою складалася з кількох ключових етапів: керування рухом, обробка даних сенсорів, тестування.

Керування рухом. Було створено алгоритми, які враховують кінематику і динаміку платформи. Це включало планування траєкторій та уникнення перешкод у різних сценаріях, наприклад, у вузьких проходах або серед рухомих об'єктів.

Обробка даних сенсорів. Розроблені модулі дозволяли отримувати та обробляти дані з камер і лідарів. Це забезпечувало платформі можливість

оцінювати своє оточення і приймати рішення на основі актуальної інформації.

Тестування. Алгоритми спочатку тестувалися у симуляції за допомогою Robotino SIM Demo. Це дозволило перевірити їх точність і стабільність у контрольованих умовах. Після цього алгоритми переносилися на реальну платформу і проходили випробування в лабораторних умовах.

Використання цих інструментів дало змогу об'єднати моделювання, розробку та тестування в один інтегрований процес. Це забезпечило ефективність розробки і високу якість кінцевого продукту, що відповідає вимогам точності та надійності.

4.3 Порядок розробки програми для уникнення перешкод мобільним роботом

Провівши аналіз початкових умов та характеру завдання, перейдемо до опису розробки програми для мобільного робота, який здатний уникати перешкоди в симуляційному середовищі. Розглянемо основні етапи коду та логіку його роботи.

Програма складається з кількох кроків:

- а) ініціалізація програми (Init);
- б) виконання початкового руху (Step1, prog1);
- в) аналіз результатів руху:

1) якщо робот виявляє перешкоду (умова `'First > 4'`), змінюється траєкторія;

2) якщо робот не стикається з перешкодою (умова `'First < 3'`), продовжується стандартний рух;

3) час роботи програми перевіряється через змінну (`'Sec > 500'`), після чого програма повертається до початкового стану.

Цей алгоритм дозволяє роботу приймати рішення в реальному часі, коригуючи свій маршрут залежно від оточуючих умов.

На рисунку 4.1 показана блок-схема програми. Вона включає такі етапи:

- а) ініціалізація (Init): запускає основний процес;
- б) Step1 (prog1): відповідає за початковий рух;
- в) Step2 (prog2): виконується у випадку виявлення перешкоди;
- г) Step3 (prog3): обробляє ситуації, коли перешкод не виявлено.

Циклічний процес триває доти, поки не досягнуто обмеження за часом.

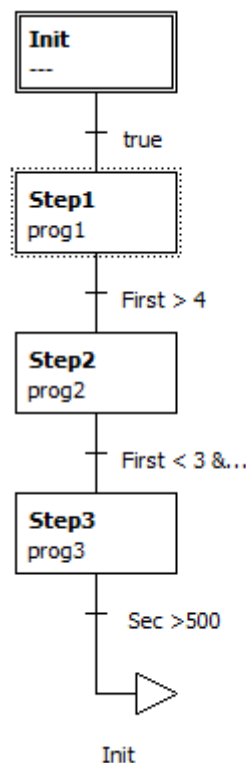


Рисунок 4.1 – Блок-схема програми

На рисунках 4.2–4.4 показано графічну реалізацію налаштувань швидкості та сенсорів у програмному середовищі Robotino SIM. Розглянемо ключові параметри:

- а) швидкість руху: встановлена на 150.0, що відповідає оптимальній швидкості для плавного маневрування;
- б) сенсори: робот використовує датчики для визначення перешкод у середовищі. Значення змінної First використовується для прийняття рішень;

в) управління Wi-Fi: забезпечує передачу даних між компонентами системи.

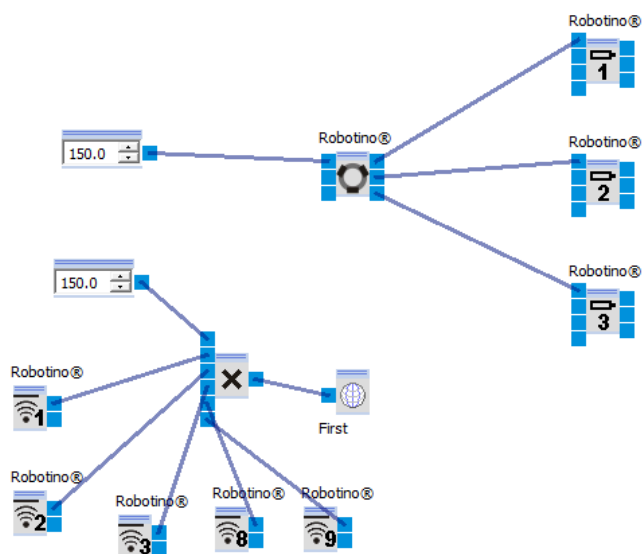


Рисунок 4.2 – Базова швидкість

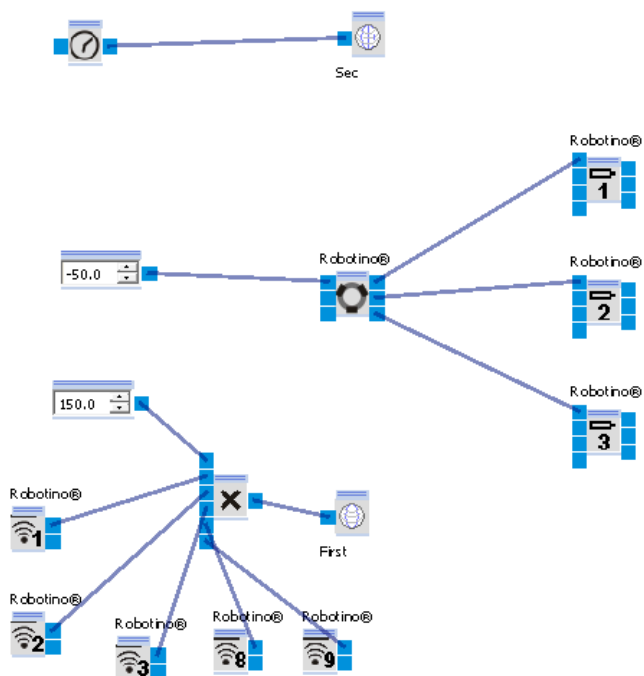


Рисунок 4.3 – Змінення швидкості для зворотного пересування

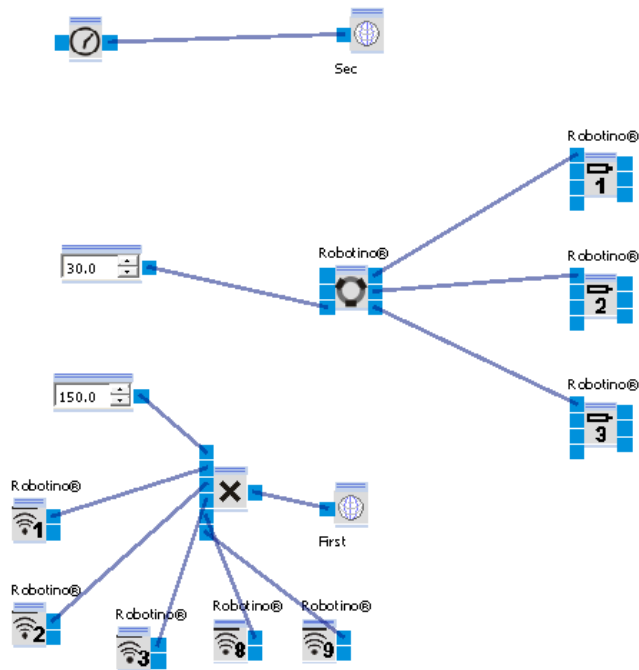


Рисунок 4.4 – Змінення куту (поворот)

Робот обладнаний системою виявлення зіткнень, яка забезпечує ефективне уникнення перешкод у процесі руху. Коли робот стикається зі стіною або іншою перешкодою, його траєкторія автоматично змінюється таким чином, щоб уникнути повторного зіткнення. Логіка роботи системи включає:

- а) реєстрацію зіткнення за допомогою сенсорів, які фіксують контакт із перешкодою;
- б) внесення змін до маршруту з урахуванням розташування попередніх перешкод для оптимального обходу;
- в) оновлення внутрішніх змінних для запобігання повторному поверненню до тієї ж перешкоди, що гарантує ефективність алгоритму.

На рисунках 4.5 та 4.6 зображено процес адаптації роботом свого маршруту після виявлення перешкоди. Основні особливості цієї системи полягають у її здатності швидко реагувати на зміни у середовищі, адаптуватися до нових умов і забезпечувати плавний рух без зупинок.

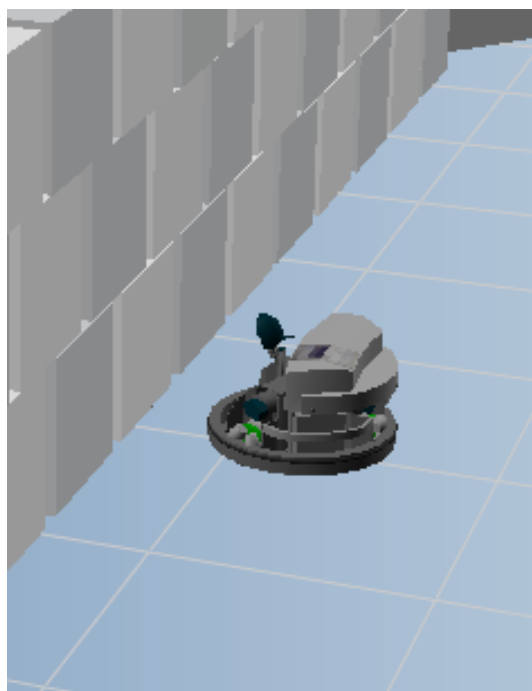


Рисунок 4.5 – Робот натикається на перешкоду

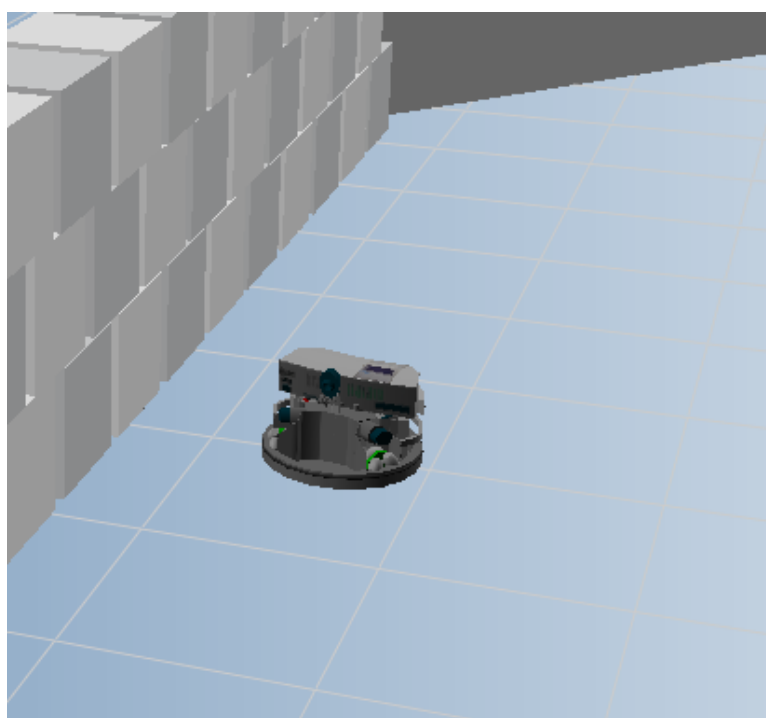


Рисунок 4.6 – Робот змінює траєкторію

Робот рухається по середовищу, дотримуючись заданих параметрів. Після зіткнення з перешкодою робот змінює кут руху та продовжує свій маршрут в обхід небезпечної зони.

4.4 Висновки до розділу

У четвертому розділі було детально розглянуто процес реалізації програмного забезпечення для управління роботизованою платформою. Основні результати роботи можна підсумувати наступним чином:

Середовище розробки проекту:

- Використання програмних інструментів Robotino SIM Demo та Robotino View 2 дозволило забезпечити інтегрований підхід до моделювання, розробки та тестування алгоритмів як у симуляційних умовах, так і на реальному обладнанні;

- Завдяки функціоналу цих середовищ стало можливим гнучко налаштовувати параметри роботи платформи, інтегрувати дані сенсорів та перевіряти алгоритми у різних сценаріях.

Реалізація алгоритмів:

- Розроблено та впроваджено алгоритми керування рухом, обробки даних сенсорів та уникнення перешкод;

- Алгоритми продемонстрували високу точність і стабільність роботи як у симуляційному середовищі, так і під час тестувань на реальній платформі;

- Підхід включав аналіз і адаптацію параметрів під реальні умови, що забезпечує надійність функціонування системи.

Розробка програми для уникнення перешкод:

- Створено програму, яка дозволяє роботу автономно уникати перешкод за допомогою сенсорів і алгоритмів прийняття рішень у реальному часі;

- Запропонована блок-схема алгоритму демонструє послідовність дій робота, починаючи з ініціалізації і завершуючи коригуванням траєкторії після виявлення перешкод;

- Проведено налаштування основних параметрів, таких як швидкість, чутливість сенсорів і маршрути руху, що сприяло ефективному виконанню завдань.

Особливості системи уникнення перешкод:

- Реалізована система дозволяє роботу швидко адаптуватися до змін середовища, вносити корективи у маршрут і забезпечувати безперервний рух;
- Завдяки використанню сенсорів та логіки оновлення даних, система зменшує ризик повторних зіткнень і покращує загальну ефективність роботи.

Загалом, реалізація програмного забезпечення показала свою ефективність у вирішенні завдань управління рухом роботизованої платформи. Поєднання симуляційного моделювання та реальних тестувань дозволило отримати надійне, точне та адаптивне рішення, яке може бути успішно застосоване у різних сферах, включаючи промисловість, логістику та дослідницькі проекти.

5 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА У НАДЗВИЧАЙНИХ СИТУАЦІЯХ

5.1 Промислова безпека в середовищі робототехнічних досліджень

Приміщення, в якому проводиться розробка алгоритмів візуалізації руху для роботизованих платформ, працює 8 годин на добу, 5 днів на тиждень. Загальна тривалість використання обладнання становить близько 2000 годин на рік. Живлення роботизованих платформ та комп'ютерів здійснюється через трифазну систему змінного струму з напругою 220 В і частотою 50 Гц.

Приміщення класифікується як таке, що не має підвищеної небезпеки ураження електричним струмом. Проте, для забезпечення безпеки роботи необхідно дотримуватись стандартів електробезпеки, зокрема:

а) встановлення заземлення для всіх робочих місць відповідно до ДБН В.2.5-27-2006;

б) регулярна перевірка опору ізоляції кабелів, яка має бути не менше 500 кОм;

в) установка пристрою захисного відключення (ПЗВ) на основних вузлах живлення для мінімізації ризику ураження електричним струмом.

Також персонал, який займається розробкою, тестуванням і обслуговуванням роботизованих платформ, повинен пройти інструктажі з охорони праці та безпеки. Види інструктажів включають:

а) вступний інструктаж: ознайомлення з загальними правилами роботи;

б) первинний інструктаж на робочому місці: роз'яснення небезпечних факторів та засобів індивідуального захисту;

в) повторний інструктаж: проводиться раз на півроку;

г) позаплановий інструктаж: проводиться у разі змін у технологічному процесі або обладнанні;

д) цільовий інструктаж: застосовується для одноразових робіт, пов'язаних з аварійними ситуаціями [21].

5.2 Виробнича санітарія у лабораторії робототехніки

Робота в лабораторії належить до категорії «легка 1а» (ДСН 3.3.6.042-99), оскільки основні завдання виконуються сидячи, за комп'ютером або керуючи роботизованими платформами. Для підтримання комфортних умов у приміщенні забезпечується:

- а) кондиціонування повітря в теплий період року;
- б) опалення в холодний період для підтримання температури в межах від 18°C до 24 °C.

Освітлення в лабораторії відповідає класу III робіт з високою точністю (ДБН В.2.5-28-2006), з мінімальною освітленістю 300 лк. Нормовані рівні шуму не повинні перевищувати 50 дБА, тому використовуються звукоізоляційні матеріали для забезпечення комфортної акустики.

5.3 Безпека у надзвичайних ситуаціях

У лабораторії робототехніки можливі надзвичайні ситуації, пов'язані із замиканням у електропроводці чи перегрівом елементів обладнання. Для мінімізації пожежних ризиків передбачено:

- а) встановлення автоматичної пожежної сигналізації;
- б) наявність вуглекислотних вогнегасників у співвідношенні 1 вогнегасник на 3 одиниці обладнання;
- в) використання кабелів із вогнезахисним покриттям.

Евакуація у разі пожежі має бути забезпечена через основний вихід із приміщення, що розташований не далі ніж за 25 метрів. План евакуації розміщений у видимому місці поблизу виходу [21].

5.4 Висновки до розділу

У п'ятому розділі було висвітлено ключові аспекти охорони праці та безпеки у надзвичайних ситуаціях у середовищі робототехнічних досліджень. Основні результати аналізу та рекомендації можна узагальнити наступним чином.

Промислова безпека:

- Приміщення для роботи з роботизованими платформами класифікується як таке, що не має підвищеної небезпеки ураження електричним струмом. Проте важливими є дотримання стандартів електробезпеки, включаючи встановлення заземлення, регулярну перевірку ізоляції кабелів і використання пристроїв захисного відключення.

- Для персоналу передбачено проведення різних видів інструктажів (вступний, первинний, повторний, позаплановий, цільовий), що сприяє підвищенню рівня обізнаності та запобіганню ризикам під час виконання робіт.

Виробнича санітарія:

- У лабораторії забезпечено належні умови для роботи відповідно до санітарних норм. Це включає підтримання оптимальної температури повітря, достатнього рівня освітлення та мінімізації шумового навантаження.

- Використання звукоізоляційних матеріалів, систем кондиціонування та опалення сприяє створенню комфортного робочого середовища, що підвищує продуктивність праці.

Безпека у надзвичайних ситуаціях:

- Для зменшення ризиків пожежі передбачено встановлення автоматичної пожежної сигналізації, використання вогнезахисних кабелів і забезпечення приміщення вуглекислотними вогнегасниками.

- План евакуації та основний вихід із приміщення відповідають нормативним вимогам, що забезпечує швидку та безпечну евакуацію у разі надзвичайної ситуації.

Заходи, описані в цьому розділі, дозволяють ефективно забезпечити безпеку працівників і обладнання під час виконання досліджень у сфері робототехніки. Виконання встановлених вимог до охорони праці та належне реагування на надзвичайні ситуації сприяють зменшенню ризиків і створенню сприятливих умов для роботи з роботизованими платформами.

ВИСНОВКИ

В ході кваліфікаційної роботи була досягнута мета підвищення ефективності роботи роботизованих платформ за рахунок впровадження системи візуалізації, яка забезпечує точне розпізнавання перешкод, визначення оптимальних траєкторій руху та адаптацію до змінних умов середовища за допомогою кінематичних та динамічних алгоритмів.

Алгоритми візуалізації руху дозволяють роботизованим платформам орієнтуватися в просторі, уникати перешкод і виконувати завдання з високою ефективністю навіть у складних і динамічних умовах. Вони включають в себе кілька основних компонентів: кінематичні алгоритми, динамічні моделі, сенсорне злиття та обробку даних у реальному часі. Кожен із цих елементів робить свій внесок у загальну продуктивність платформи.

Оптимізація алгоритмів є важливим завданням, оскільки вона визначає швидкість реакції платформи, енергоефективність і здатність працювати в обмежених апаратних умовах. У процесі розробки необхідно використовувати сучасні інструменти та підходи, такі як фільтрація даних, модульна архітектура і паралельна обробка інформації.

Популярність автономних платформ, таких як роботизовані візки для складів, дрони для доставки чи дослідницькі підводні апарати, свідчить про зростання інтересу до їх використання у різних галузях. Завдяки можливостям виконувати завдання без участі людини, такі платформи значно підвищують продуктивність процесів.

Сучасні технології, такі як машинне навчання, глибоке навчання та інтеграція з хмарними сервісами, відкривають нові можливості для вдосконалення алгоритмів візуалізації. Вони дозволяють підвищити точність роботи платформ, покращити обробку сенсорних даних і забезпечити гнучкість систем.

У ході виконання кваліфікаційної роботи було вирішено наступні завдання:

- а) проведено аналіз предметної області «Розробка алгоритмів візуалізації руху для роботизованих платформ»;
- б) сформульовано мету та завдання роботи;
- в) обрано відповідні інструменти та мови програмування для реалізації алгоритмів;
- г) розроблено алгоритми керування рухом і обробки даних сенсорів;
- д) виконано тестування алгоритмів у симуляційному середовищі;
- е) реалізовано алгоритми на реальній роботизованій платформі.

Таким чином, поставлені завдання з розробки алгоритмів візуалізації руху для роботизованих платформ, що були визначені метою роботи, виконані у повному обсязі. Отримані результати можуть бути використані для подальшого вдосконалення робототехнічних систем та впровадження їх у практичні завдання.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. ДСТУ 3008-15. Документація. Звіти у сфері науки та техніки. структура та правила оформлення. Введ. 2015-06-22. К. Держстандарт України, 2017. 29 с.
2. Дослідження алгоритмів візуалізації руху роботизованої платформи / Т. В. Ісмаїлов, М. Г. Стародубцев. Харків, 2025. 4с.
3. 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.
4. ДСТУ 8302:2015. Інформація та документація. Бібліографічне посилання. Загальні положення та правила складання. Введ. 2016-07-01. К. Держстандарт України, 2016. – 16 с.
5. Two home service robot platforms used for this work: Toyota HSR Human Support Robot [Електронний ресурс] // ResearchGate. – Режим доступу: https://www.researchgate.net/figure/Two-home-service-robot-platforms-used-for-this-work-a-Toyota-HSR-Human-Support_fig1_343871743
6. Ремонт та модернізація обладнання [Електронний ресурс] // Офіційний сайт DLMZ. – Режим доступу: <https://dlmz-ua.com/ua/remont-modernizaciya-obladnannya/>
7. Баранов С.С. Класифікація робототехнічних платформ та готових технічних рішень для навчання учнів основ робототехніки // Відкрите освітнє е-середовище сучасного університету. – 2021. – №11. – С. 1–12.
8. Медвідь А.Я., Яковина В.С. Інтеграція даних про колізії для пришвидшення оптимізації траєкторії руху роботизованої руки // Науковий вісник НЛТУ України. – 2024. – Т. 34, №5. – С. 136–143.

9. Ваше здоров'я: медична робототехніка в наші дні [Електронний ресурс] // Robotics.ua. – Режим доступу: <https://robotics.ua/vashe-zdorove-medytsynskaia-robototekhnyka-v-nashy-dny/>
10. Логістичні процеси, які можна (і потрібно) автоматизувати [Електронний ресурс] // Isitlab – Режим доступу: <https://isitlab.com/blog/logistics-processes-that-can-and-should-be-automated/>
11. Караман С., Фраццолі Е. Алгоритми на основі вибірки для оптимального планування руху // The International Journal of Robotics Research. – 2011. – Т. 30, №7. – С. 846–894.
12. Основи наукових досліджень : підручник / І. Ш. Невлюдов, Ю. М. Олександров, А. О. Андрусевич, О. О. Чала ; М-во освіти і науки України, Харків. нац. ун-т радіоелектроніки. – Prague : OKTAN PRINT, 2024. – 468 с.
13. Анісімова Т.І., Сабірова Ф.М., Шатунова О.В. Формування проєктних та дослідницьких компетентностей у майбутніх вчителів у рамках STEAM-освіти // International Journal of Emerging Technologies in Learning (iJET). – 2020. – Т. 15, №2. – С. 204–217.
14. Tesla Model S [Електронний ресурс] // Офіційний сайт Tesla. – Режим доступу: <https://www.tesla.com/models>.
15. Pepper: The humanoid robot [Електронний ресурс] // SoftBank Robotics. – Режим доступу: <https://corporate-internal-prod.aldebaran.com/en/pepper>
16. Analysis flow chart of GLR algorithm [Електронний ресурс] // ResearchGate. – Режим доступу: https://www.researchgate.net/figure/Analysis-flow-chart-of-GLR-algorithm_fig2_361635914
17. Роботи на складах: приклади автоматизації [Електронний ресурс] // Wareteka. – Режим доступу: <https://wareteka.com.ua/uk/blog/roboti-na-skladah-prikladi-avtomatizaciyi/>
18. ЛаВалле С.М. Швидко-зростаючі випадкові дерева: новий інструмент для планування шляхів // 1998.

19. Кім Д., Квон Ю., Юн С.-Е. Адаптивна лінива перевірка колізій для оптимального планування руху на основі вибірки // 2018 15th International Conference on Ubiquitous Robots (UR). – 2018. – С. 320–327.

20. Лю Ч., Ву Л., Сяо В., Лі Г., Сюй Д., Го Дж., Лі В. Покращений алгоритм оптимізації колонії мурах для вирішення задачі планування шляхів // Knowledge-Based Systems. – 2023. – Т. 271.

21. Методичні вказівки до самостійної роботи з дисципліни "Охорона праці в галузі" для студентів усіх спеціальностей галузі автоматизації та приладобудування денної форми навчання / упоряд.: Г. В. Пронюк, Н. М. Сердюк, Т. Є. Стиценко ; М-во освіти і науки України, ХНУРЕ. – Харків : ХНУРЕ, 2016. – 116 с.