

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Автоматики і комп'ютеризованих технологій
(повна назва)

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

АТЕСТАЦІЙНА РОБОТА

Пояснювальна записка

другий (магістерський)

(рівень вищої освіти)

Удосконалення методу прокладення шляху мобільної платформи в
невизначеному просторі

(тема)

Виконав: студентка 2 курсу, гр. КІТПВм-19-1
Мамонько Д.В.

(прізвище, ініціали)

Спеціальність 151 Автоматизація
та комп'ютерно-інтегровані технології
освітньої програми Комп'ютерно-інтегровані
технологічні процеси і виробництва

(код і повна назва напрямку)

Тип програми освітньо-професійна
(повна назва освітньої програми)

Керівник проф. Невлюдов І.Ш.
(посада, прізвище, ініціали)

Допускається до захисту
зав. кафедри

(підпис)

Невлюдов І.Ш.

(прізвище, ініціали)

2020 р.

Харківський національний університет радіоелектроніки

Факультет	Автоматики і комп'ютеризованих технологій
Кафедра	Комп'ютерно-інтегрованих технологій, автоматизації та мехатроніки
Рівень вищої освіти	другий (магістерський)
Спеціальність	151 Автоматизація та комп'ютерно-інтегровані технології
Тип програми	освітньо-професійна
Освітня програма	Комп'ютерно-інтегровані технологічні процеси і виробництва
(код і повна назва)	

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)
«_____» _____ 2020 р.

ЗАВДАННЯ
НА АТЕСТАЦІЙНУ РОБОТУ

студентові	Мамонько Даші Вячеславівні
(прізвище, ім'я, по батькові)	
1. Тема роботи	Удосконалення методу прокладення шляху мобільної платформи в невизначеному просторі
затверджена наказом по університету від 02.11. 2020 р. № 1511 Ст	
2. Термін подання студентом роботи до екзаменаційної комісії	10.12.2020 р.
3. Вихідні дані до роботи	
3.1 Використання методу пошуку оптимальної траєкторії руху та методу візуальної одометрії;	
3.2 Мова програмування C#;	
3.3 Середовище розробки Visual Studio;	
3.4 Використання імітаційного (програмного) моделювання для проведення експериментальних досліджень	
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)	
4.1 Вступ	
4.2 Аналіз існуючих принципів локалізації мобільних транспортних засобів	
4.3 Аналіз методів прокладення шляху в невизначеному просторі	
4.4 Дослідно-аналітичний розділ	
4.5 Експериментальні дослідження	
4.6 Висновки та перелік використаних джерел	

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) Демонстраційний матеріал представлений у форматі презентації PowerPoint (*.ppt) – 23 с. формату А4

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

Найменування розділу	Керівник (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Основна частина	Невлюдов І.Ш.		

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Провести аналіз літератури за темою роботи	06.11.2020	Виконано
2	Провести аналіз методів прокладення шляху в невизначеному просторі	10.11.2020	Виконано
3	Провести розробку алгоритму вирішення задачі пошуку шляху переміщення	13.11.2020	Виконано
4	Розробити програму для проведення моделювання	18.11.2020	Виконано
5	Провести експериментальні дослідження	27.11.2020	Виконано
6	Провести розрахунки, пов'язані з охороною праці	30.11.2020	Виконано
7	Оформлення пояснювальної записки	03.12.2020	Виконано
8	Подання роботи до ЕК	10.12.2020	Виконано

Дата видачі завдання 02.11.2020 року

Студент

(підпис)

Керівник роботи

(підпис)

Мамонько Д.В.

(прізвище, ініціали)

проф. Невлюдов І.Ш.

(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 78 с., 33 рис., 1 дод., 29 джерел.

AMCL, LIDAR, SLAM, АЛГОРИТМ, ВІЗУАЛЬНА ОДОМЕТРІЯ,
ДИНАМІЧНІ ПЕРЕШКОДИ, ІМІТАЦІЙНЕ МОДЕЛЮВАННЯ,
ПОБУДОВИ КАРТИ, СЕНСОРИ, ТРАЄКТОРІЯ.

Об'єктом дослідження – інтелектуальне роботизоване виробництво.

Предмет дослідження – методи оптимізації траєкторії руху мобільного транспортного засобу в невизначеному просторі.

Мета роботи – визначення методу оптимізації траєкторію переміщення транспортного робота у виробничому цеху серед неоднорідних перешкод для значного зменшення тривалості транспортних операцій.

У роботі проведено:

- дослідження існуючих методів визначення траєкторії переміщення мобільного робота в виробничому середовищі серед перешкод;
- виконано аналіз існуючих принципів локалізації мобільних транспортних засобів в просторі та одночасного прокладення траєкторії руху;
- обрано програмну платформу для реалізації визначеного методу прокладення шляху;
- проведено експериментальне дослідження сумісного використання методу локалізації та прокладення шляху в умовах невизначених перешкод.

Результати магістерської атестаційної роботи апробовані у 4 публікаціях (3 статті та 1 тези доповідей).

ABSTRACT

Explanatory note: 78 p., 33 fig., 1 app., 29 sources.

AMCL, LIDAR, SLAM, ALGORITHM, VISUAL ODOMETRY, DYNAMIC INTERFERENCE, SIMULATION SIMULATION, MAP CONSTRUCTIONS, SENSORS, TRAJECTORY.

The object of research is intelligent robotic production.

The subject of research - methods of optimizing the trajectory of a mobile vehicle in an indefinite space.

The purpose of the work is to determine the method of optimizing the trajectory of the transport robot in the production shop among the inhomogeneous obstacles to significantly reduce the duration of transport operations.

The work carried out:

- study of existing methods for determining the trajectory of the mobile robot in the production environment among the obstacles;
- the analysis of the existing principles of localization of mobile vehicles in space and simultaneous laying of a trajectory of movement is executed;
- a software platform was chosen to implement a certain method of paving the way;
- an experimental study of the joint use of the method of localization and paving in conditions of uncertain obstacles.

The results of the master's attestation work were tested in 4 publications (3 articles and 1 abstracts).

ЗМІСТ

Скорочення та умовні позначки.....	8
Вступ.....	9
1 Аналіз існуючих принципів локалізації мобільних транспортних засобів в просторі та одночасного прокладення траєкторії руху.....	11
1.1 Аналіз існуючих методів навігації	11
1.1.1 Методи позиціонування.....	11
1.1.2 Візуальна одометрія.....	13
1.2 Опис алгоритму візуальної одометрії.....	15
1.3 Аналіз принципу роботи мобільного робота, що використовується в роботі.....	19
1.4 Аналіз структурної схеми мобільного робота.....	25
1.5 Постановка задач.....	27
1.6 Висновки по першому розділу.....	27
2 Методи прокладення шляху в невизначеному просторі.....	28
2.1 Методи побудови карти простору	28
2.2 Методи прокладення шляху в просторі на основі алгоритму A*.....	32
2.3 Висновки по другому розділу.....	37
3 Дослідно-аналітичний розділ.....	39
3.1 Опис запропонованого методу визначення траєкторії руху в невизначеному просторі.....	39
3.2 Вибір методу візуальної одометрії.....	42
3.3 Розробка алгоритму вирішення задачі пошуку шляху переміщення.....	46
3.4 Висновки по третьому розділу роботи.....	49
4 Експериментальні дослідження.....	50
4.1 Опис методики проведення експериментальних досліджень.....	50

4.2 Опис структури програми для проведення моделювання.....	52
4.3 Тестовий запуск програми та проведення іспитів.....	58
4.4 Розрахунок загальнообмінної вентиляції приміщення в якому проводиться розробка.....	69
4.5 Висновки по четвертому розділу роботи.....	71
Висновки.....	73
Перелік джерел посилань.....	75
Додаток А Демонстраційний матеріал.....	79

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

AMCL – Adaptive Monte Carlo Localization;

GPS – Global positioning system;

IMU – inertial measurement unit;

LIDAR – Light Detection and Ranging;

PWM – широко-імпульсна модуляція;

SAD – sum of absolute difference;

SLAM – simultaneous localization and mapping;

ДП – Державне підприємство;

ДСТУ – Державний стандарт України;

МП – мобільна платформа

ПЗЗ-матриця – прилад із зарядним зв'язком.

ВСТУП

При впровадженні у виробництво концепції Industry 4.0 особливе значення має використання мобільних роботів для поєднання промислового обладнання в один цілісний виробничий процес.

Задача сумісного використання роботів та людини є дуже складною. По-перше, треба забезпечити прокладення шляху переміщення мобільних транспортних засобів серед об'єктів, положення яких динамічно змінюються. По-друге, треба забезпечити безпеку персоналу, що обслуговує обладнання за рахунок використання інтелектуальних сенсорів.

Таким чином, тема атестаційної роботи «Удосконалення методики прокладення шляху мобільної платформи в невизначеному просторі» є актуальною. Актуальність цієї теми обумовлена необхідністю оптимізації траєкторії пересування мобільних транспортних роботів при перевезенні корисного навантаження (деталей, складальних одиниць) у виробничому цеху серед неоднорідних перешкод між промисловим обладнанням, з однієї ділянки на іншу, або між робочими місцями.

Мета роботи – визначення методу оптимізації траєкторію переміщення транспортного робота у виробничому цеху серед неоднорідних перешкод для значного зменшення тривалості транспортних операцій.

Вирішивши поставлену мету буде забезпечено застосування найкоротшого шляху проходження усіх стадій і операцій виробничого процесу, від запуску у виробництво початкових матеріалів, до виходу готової продукції.

Об'єктом дослідження – інтелектуальне роботизоване виробництво.

Предмет дослідження – методи оптимізації траєкторії руху мобільного транспортного засобу в невизначеному просторі.

Для досягнення поставленої мети необхідно вирішити наступні завдання:

- дослідити існуючі методи визначення траєкторії переміщення мобільного робота в виробничому середовищі серед перешкод;
- виконати аналіз існуючих принципів локалізації мобільних транспортних засобів в просторі та одночасного прокладення траєкторії руху;
- обрати програмну платформу для реалізації визначеного методу прокладення шляху;
- провести експериментальне дослідження сумісного використання методу локалізації та прокладення шляху в умовах невизначених перешкод;
- оформити атестаційну роботу магістра згідно ДСТУ 3008:2015 [1] та методичними вказівками з «Розробки й оформлення магістерської атестаційної роботи» для студентів другого (магістерського) рівня вищої освіти галузі знань 15 Автоматизація та приладобудування за спеціальністю 151 Автоматизація та комп'ютерно-інтегровані технології освітні програми: «Автоматизоване управління технологічними процесами», «Комп'ютерно-інтегровані технологічні процеси і виробництва», «Комп'ютеризовані та робототехнічні системи» [2].

1 АНАЛІЗ ІСНУЮЧИХ ПРИНЦИПІВ ЛОКАЛІЗАЦІЇ МОБІЛЬНИХ ТРАНСПОРТНИХ ЗАСОБІВ В ПРОСТОРІ ТА ОДНОЧАСНОГО ПРОКЛАДЕННЯ ТРАЄКТОРІЇ РУХУ

1.1 Аналіз існуючих методів навігації

1.1.1 Методи позиціонування

Системи позиціонування забезпечують визначення місцезнаходження мобільної платформи з прив'язкою до карти місцевості або об'єкту. Існують системи абсолютного (глобального) і відносного позиціонування.

Абсолютне позиціонування має на увазі отримання кожного разу нових координат, за допомогою супутникової навігації, незалежно від попереднього місця розташування.

Система супутникової навігації включає орбітальне угруповання, що складається з декількох (від 2 до 30) супутників, випромінюючих спеціальні радіосигнали, наземну систему контролю, приймальне клієнтське устаткування. Можливо також використання системи наземних станцій, що дозволяють підвищити точність визначення координат, системи для передачі користувачам уточнених координат і годинника супутників.

Нині існує декілька систем супутникової навігації [3]:

- GPS (Global positioning system) – система супутникової навігації, що належить міністерству оборони США. GPS дозволяє у будь-якому місці Землі (за винятком приполярних областей), майже за будь-якої погоди, а також в космічному просторі, поблизу планети визначити місце розташування і швидкість об'єктів.

- Бэйдоу – китайська система супутникової навігації, знаходиться на стадії запуску супутників. Особливість – невелика кількість супутників, що знаходяться на геостаціонарній орбіті.

– GALILEO – система супутникової навігації Європейського союзу, що знаходиться на етапі створення супутникового угруповання.

Кожна з цих систем транслює одні і ті ж дані на двох частотах. Це робиться для того, щоб вичислити іоносферну затримку, оскільки іоносферна затримка залежить від частоти і ця залежність відома.

У кожній області, де використовується та або інша система супутникової навігації, існують свої вимоги до точності знаходження місця розташування. Відповідно до цих вимог використовуються різні датчики і різні методи.

Системи локального позиціонування знаходять широке застосування в сільськогосподарських і промислових галузях, що обумовлено їх високою точністю визначення координат об'єктів (1-3 м) і можливістю забезпечення безперервного відстежування місця розташування об'єктів як зовні, так і усередині будівель.

Системи локального позиціонування можуть бути двох видів:

– автономними – повна незалежність методики визначення координат від наявності зовнішніх випромінюваних сигналів;

– неавтономні – призначений для користувача пристрій вимірює затримку сигналу, що приходить від розташованих навколо базових станцій.

Найбільш поширені системи навігації використовують дані с сенсорів, що встановлені на роботі, для визначення відносного положення пристрою в локальному просторі. Таким сенсорами можуть бути:

– оптичні одометри – оптичні датчики, що розташовані на вісі колеса робота та видають певну кількість імпульсів за одне обертання;

– цифрові камери, лазерні сканери – оптичні пристрої на основі ПЗЗ-матриці, що дозволяють отримати інформацію про певну ділянку простору (оптична одометрія).

На рисунку 1.1 показано принцип встановлення камери на роботі для отримання візуальних даних про середовище попереду мобільного пристрою.

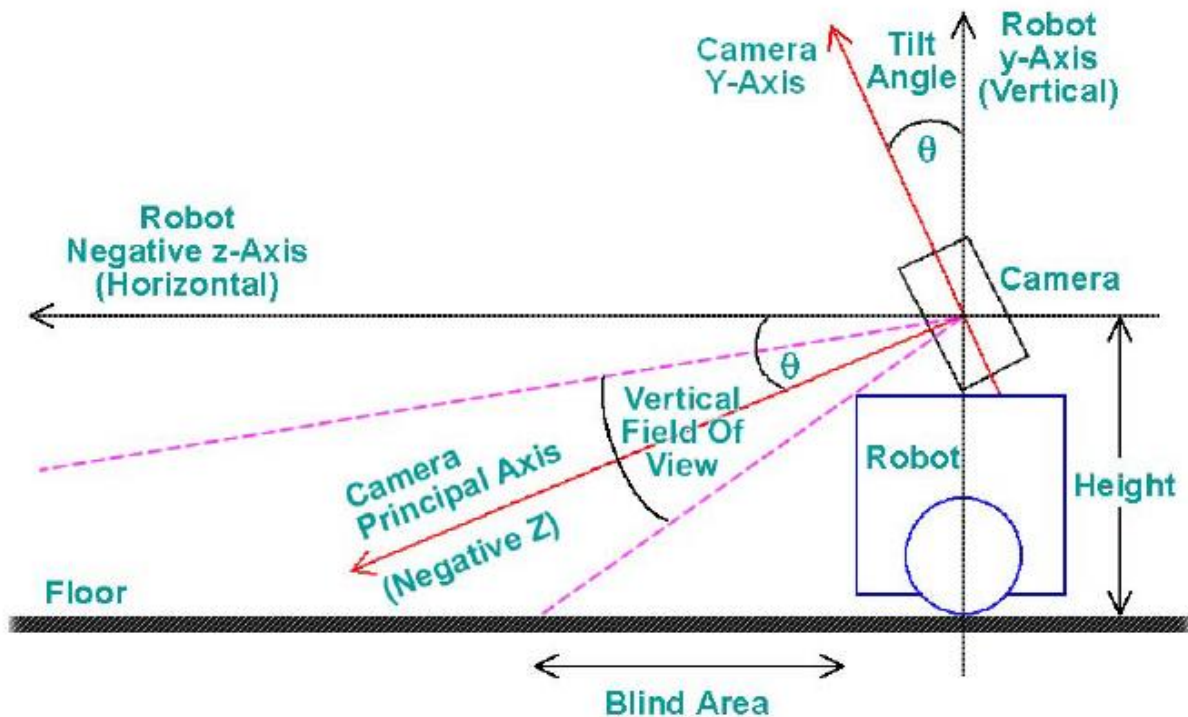


Рисунок 1.1 – Принцип встановлення камери на роботі для отримання візуальних даних про середовище попереду мобільного пристрою [4]

1.1.2 Візуальна одометрія

У робототехніці і комп'ютерному баченні, візуальна одометрія – процес визначення позиції і орієнтації робота шляхом аналізу послідовних зображень отриманих за допомогою камери.

У навігації, одометрія це процес використання даних про навколишнє середовище і датчиків руху для визначення змін позиції робота, для цього використовуються такі пристрої як датчики кута повороту для виміру оборотів коліс. Хоча такий метод корисний для багатьох колісних або гусеничних транспортних засобів, традиційні методи одометрії не можна застосувати для портативних роботів з нестандартними методами пересування, наприклад роботів, що рухаються завдяки ногам. Крім того, в методі одометрії є проблеми з точністю, оскільки колеса можуть проскользувати і тим самим вносити погрішність в оцінку пройденої відстані. Помилка збільшується коли транспортний засіб рухається по нерівних поверхнях. Помилки мають тенденцію накопичуватися з часом.

Візуальна одометрія є процесом визначення подібної інформації для оцінки відстані переміщення на основі послідовних зображень отриманих з камери. Візуальна одометрія дозволяє підвищити точність навігаційних приладів в роботах або транспортних засобах, які використовують будь-який спосіб пересування відносно поверхні.

Цей метод визначення положення в просторі оснований на послідовному обчисленні переміщення робота на основі зображень з камери. Обчислення проводяться між новими і попередніми кадрами і об'єднані ці переміщенні для отримання траєкторії руху об'єкту в реальному часі [5].

Більшість з існуючих підходів у візуальній одометрії мають в основі наступні кроки.

а) є ряд вхідних зображень знятих з використанням однієї камери, стерео камери, або всенаправленої камери;

б) корекція зображень, застосовується техніка обробки зображень для усунення спотворень об'єктиву і т.д.;

в) виявлення ознак: визначити оператори пошуку і знайти відповідності ознак по кадрах і побудувати поле оптичного потоку:

1) використовується кореляція для встановлення відповідності двох зображень, і не довгострокових відстежування ознак;

2) виділення ознак і кореляція;

3) побудова поля оптичного потоку (метод Лукаса – Канаді);

г) перевірка векторів оптичного потоку на наявність потенційних помилок і видалення відхилень;

д) визначення руху камери з оптичного потоку:

1) варіант 1: Фільтр Кальмана для оцінки розподілу;

2) варіант 2: знайти геометричні і 3d властивостей об'єктів, які мінімізують функцію втрат, ґрунтована на оцінці помилки повторної проєкції між двома сусідніми зображеннями. Це можна здійснити за допомогою математичної мінімізації або випадкового відбору;

е) періодична репопуляції точок шляху для підтримки відповідності шляху уздовж зображення.

Альтернативою методу виділення ознак являється "прямий" метод, який мінімізує помилку напряму в просторі зображення без попереднього пошуку ознак і їх порівняння.

Інший метод, визначає площинні переміщення і повороти між зображеннями використовуючи метод фазової кореляції замість виділення ознак.

1.2 Опис алгоритму візуальної одометрії

Загальний алгоритм візуальної одометрії з виглядає так:

а) підготовка даних (фреймів RGB і глибини), далі, коли згадується поточний фрейм – мається на увазі дані з кадру глибини і RGB-фрейма, отримані на поточній ітерації;

б) пошук характерних особливостей на поточному кадрі (FAST) – характерні особливості це, наприклад, кути;

в) сегментація – зображення розбивається на прямокутні ділянки, в кожному з яких відбирається задане число самих яскраво виражених характерних особливостей;

г) визначаємо зміщення поточного фрейма відносно попереднього;

д) на основі знайденого зміщення здійснюємо пошук і порівняння характерних особливостей не відразу в усьому зображенні, а по ділянках, що відповідають другу-другу, для порівняння і вирівнювання використовується декілька методів:

1) значення SAD (sum of absolute difference) – абсолютна сума відмінностей між двома особливостями + вирівнювання на основі методу найменших квадратів

2) будується граф, де кожна вершина відповідає парі особливостей, що співпала. Ребра проводяться між вершинами у випадку якщо відстань між

особливостями не перевершує заданого порогового значення, упродовж аналізу усієї послідовності. (Якщо на двох кадрах ми знайшли декілька особливостей – наприклад кути будівлі – відстань Евкліда між ними буде постійною від кадру до кадру). У цьому графові шукається максимальний зв'язний підграф – тобто усі вершини якого сполучені ребрами;

е) фінальне вирівнювання результатів:

- 1) мінімізації дистанції Евкліда між особливостями, що співпали;
- 2) мінімізації помилки проектування (з попереднього кадру на поточний і навпаки);
- 3) мінімізації масштабного відхилення за допомогою ключових кадрів;

є) доповнення загальної сцени даними з поточного сцени.

В результаті кроків 1-7 ми отримуємо фінальну ізометрію, досить точну для подальшого конструювання глобальної сцени і позиціонування камери в ній.

На рисунку 1.2 показано принцип отримання зображення з моно-камери.

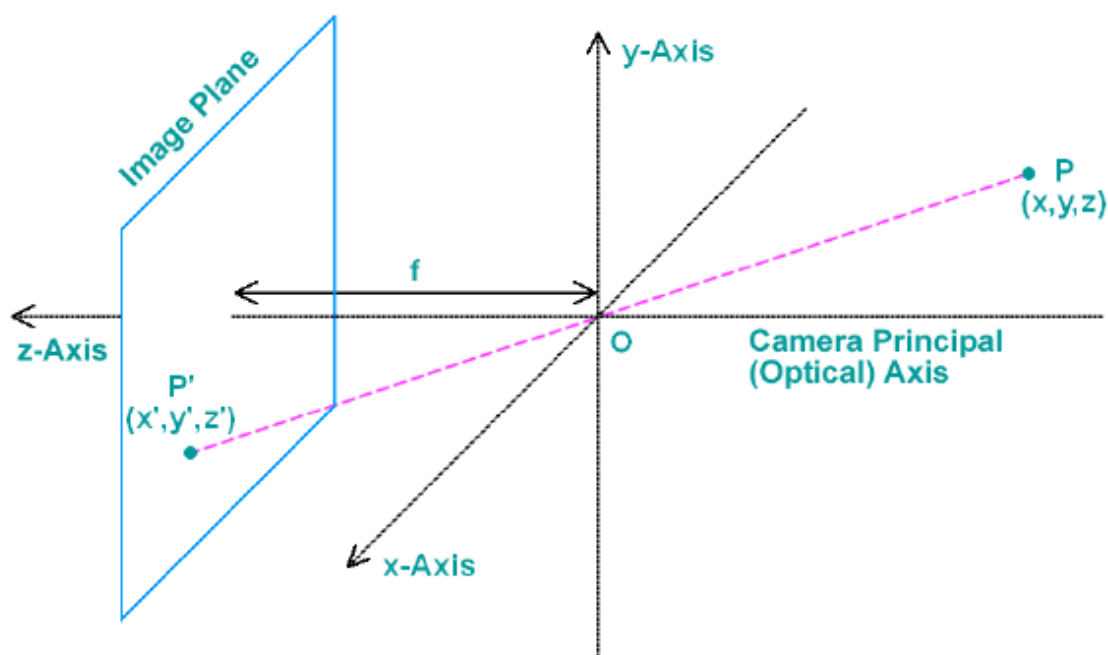


Рисунок 1.2 – Принцип отримання зображення з моно-камери

На рисунку 1.2 об'єктив камери знаходиться у початку системи координат камери, позначеної O . Світловий промінь із точки в реальному світі $P(x, y, z)$ проходить крізь отвір і потрапляє в площину зображення в точці $P'(x', y', z')$.

Зображення створюється на площині під прямим кутом до головної (оптичної) осі камери. Щоб підтримувати правильну систему координат, ось z спрямована від отвору камери (початку координат) до площини зображення. Тому відстань вздовж осі z до площини зображення залишається постійною і відображається як f на діаграмі.

Рівняння, за допомогою якого виконується проекція координати P' до P , можна отримати за допомогою відомих геометричних перетворювань [6]:

$$\frac{x'}{f} = \frac{x}{z}, \quad (1.1)$$

$$\frac{y'}{f} = \frac{y}{z}. \quad (1.2)$$

Якщо об'єктив у камері ідеальний, то застосовуються ті самі рівняння. Однак справжні лінзи не є ідеальними, тому виникає певна кількість спотворень, що ускладнює рівняння.

У комп'ютерному зорі компенсація спотворення зображення у вигляді перспективи визначається як проектне відображення точок однієї площини на іншу.

Наприклад, таким перетворенням буде проектування точок плоского зображення маркера, розташованого в тривимірному просторі, на площину зображення камери (рис. 1.3).

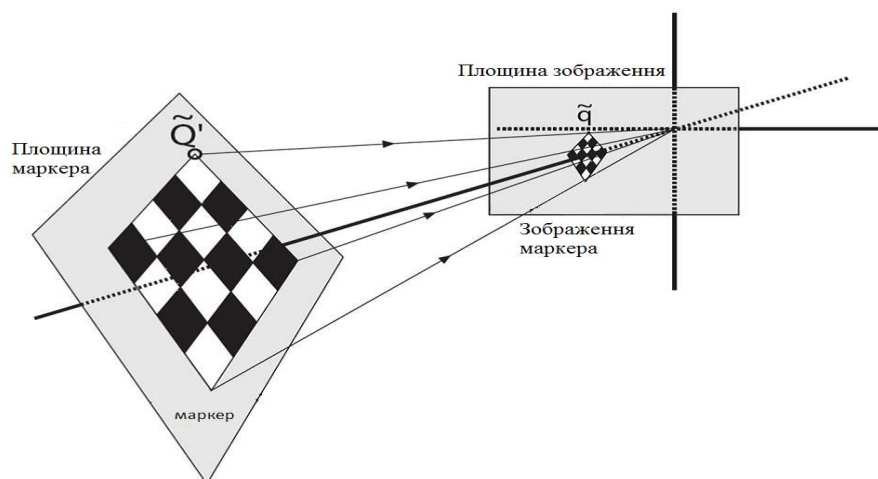


Рисунок 1.3 – Проектне відображення точок однієї площини на іншу

Це відображення можна описати у вигляді матриці H такий, що

$$\tilde{q} = sH\tilde{Q}, \quad (1.3)$$

$$\tilde{Q} = [X \ Y \ Z \ 1]^T, \quad (1.4)$$

$$\tilde{q} = [x \ y \ z \ 1]^T. \quad (1.5)$$

де s – коефіцієнт масштабування. Він є частиною перетворення, але його зазвичай явно виносять як множник перед матрицею H .

Гомографічне перетворення включає фізичне перетворення і проектування на площину зображення.

Фізичне перетворення складається з обертання R і переміщення t . Оскільки ми працюємо в однорідних координатах, ми можемо записати ці перетворення у вигляді однієї матриці [7]

$$W = [R \ t] \quad (1.6)$$

Для операції проектування використовуються введені раніше внутрішні параметри камери M

$$M = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}. \quad (1.7)$$

Тоді перетворення має вигляд

$$\tilde{q} = sMW\tilde{Q}. \quad (1.8)$$

Розглянемо точку $\tilde{Q}$, яка представляє точку $\tilde{Q}$ на площині маркера.

Остаточно, перетворення набирає вигляду:

$$q = sH Q, \quad (1.9)$$

$$H = M[r_1 \quad r_2 \quad t]. \quad (1.10)$$

де r_1, r_2 – це стовпці матриці повороту R , і матриця H має розмірність 3×3 .

1.3 Аналіз принципу роботи мобільного робота, що використовується в роботі

Для побудови шляху переміщення робот повинен знати, де він знаходиться в кожний момент часу, з високим ступенем точності. Позиція роботу складається з двовимірної координати положення в просторі та його орієнтації. Таким чином у нас використовується три змінні: x, y, A .

В нашому дослідженні ми використовували типову конструкцію мобільного робота з диференціальним приводом. Такі пристрої часто використовуються як у наукових дослідженнях, так і в промисловості.

На рисунку 1.4 показано зовнішній вигляд роботи.

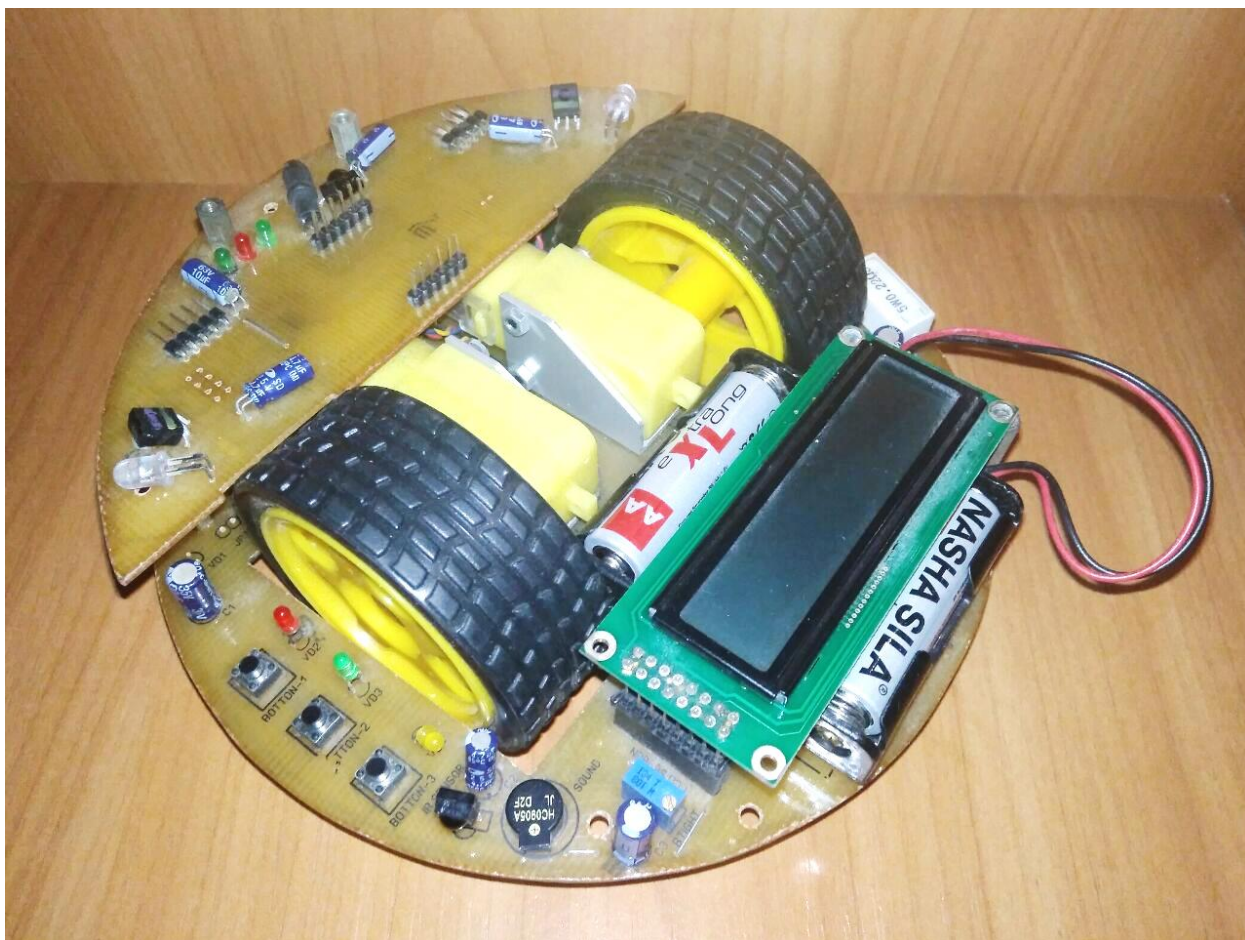


Рисунок 1.4 – Зовнішній вигляд роботи

Як можна бачити з наведеного рисунку, пристрій складається з двох коліс, якими можна керувати незалежно, вмикаючи привід як вперед, так і в зворотному напрямку. Також в конструкції використовується опорне колесо для балансу. Це колесо ми не беремо до розрахунку при визначенні положення роботи в просторі, тому що воно не впливає на зміну руху.

Така конфігурація роботи має назву «Неголономна», оскільки робот не може миттєво рухатись у будь-якому напрямку але система керування рухом, дозволяє йому досягти будь-якої точки в 2D просторі.

На рисунку 1.5 показана схема роботи з двома диференціальним приводом та двома колесами.

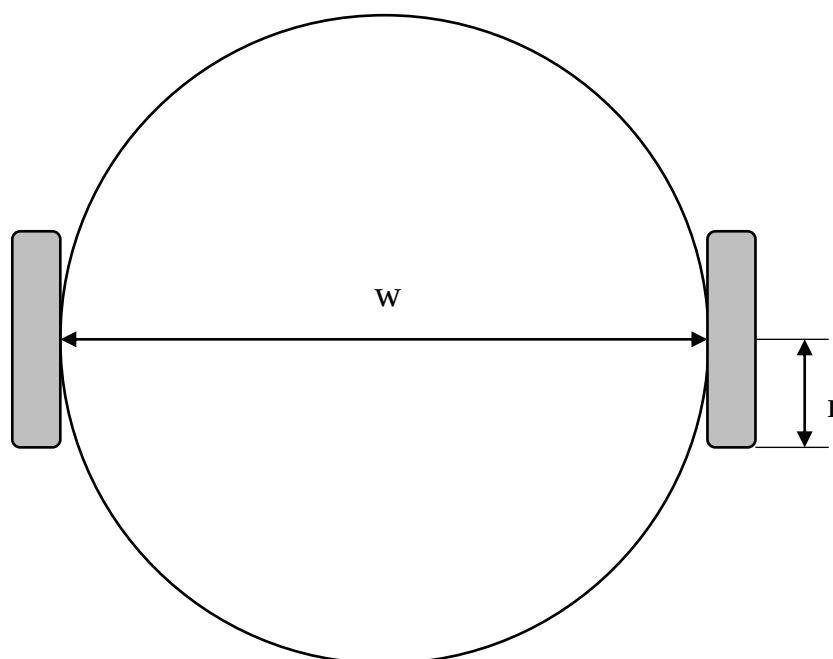


Рисунок 1.5 – Схема роботи з двома диференційним приводом та двома колесами

Тип роботів, що показано на рисунку 1.5 може їздити вперед або назад, обертатися на місці або їздити довільними дугами.

Якщо рух робота обмежуються лише прямолінійним переміщенням або обертаннями, то рівняння руху пов'язане з геометрією робота та роздільною здатністю колісних енкодерів. Для розрахунку повинні бути відомі три параметри (рис. 1.5):

- кількість імпульсів колісних енкодерів за один оборот колеса, n ;
- радіус коліс, r ;
- колісна база або відстань між колесами, w .

Для руху вперед обидва колеса повинні обертатися на однакову величину. Через спосіб встановлення двигунів, як правило, це призводить до того, що значення датчика коліс змінюються в протилежних напрямках, але обидва значення повинні змінюватися на однакову абсолютну величину, Δe .

Отже, пройдена відстань d становить:

$$d = 2\pi r \cdot \frac{\Delta e}{n}, \quad (1.11)$$

де r – радіус коліс;

Δe – кількість імпульсів енкодера;

n – кількість імпульсів колісних енкодерів за один оборот колеса.

Коли робот обертається на місці, колеса рухаються в протилежних напрямках по колу діаметром, що дорівнює колісній базі, w . (рис. 1.5). У цьому випадку кут повороту, α вимірюється відносно центру роботу:

$$\alpha = \frac{4\pi r}{w} \cdot \frac{\Delta e}{n}, \quad (1.12)$$

де w – колісна база роботу;

r – радіус коліс;

Δe – кількість імпульсів енкодера;

n – кількість імпульсів колісних енкодерів за один оборот колеса.

З формули 1.12 можна отримати потрібну кількість імпульсів для здійснення необхідного типу руху.

Зазвичай, мобільними роботами керують, регулюючи швидкість обертання коліс. Іноді це просто називають швидкістю робота, але швидкість обертання колеса та швидкість робота (швидкість осі) не однакові. При такому підході робот постійно рухається, і тому йому потрібно приймати рішення в режимі реального часу, щоб уникнути зіткнень та дослідити своє оточення.

Важливою якістю електродвигуна постійного струму є те, що можна змінити напрям обертання шляхом перемикання полярності прикладеної напруги. Це досягається за допомогою Н-моста, що дозволяє двигуну

обертатися вперед або назад. На рисунку 1.6 показано принцип роботи Н-мосту [8].

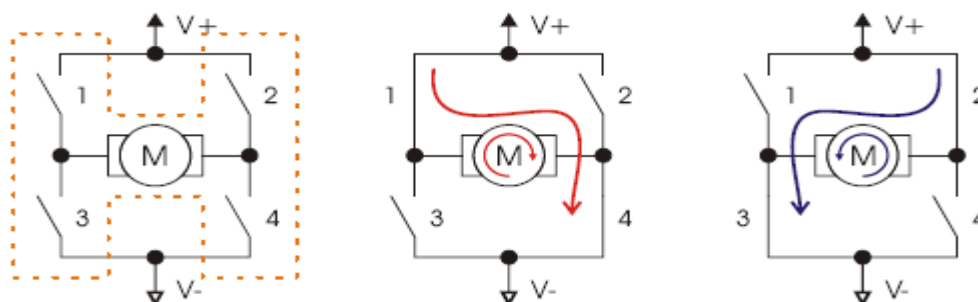


Рисунок 1.6 – Принцип роботи Н-мосту

Якщо перемикачі 1 і 4 закриті (у центрі зображення), струм через двигун йде зліва направо, і двигун розкручується вперед. Закриття перемикачі 2 і 3 викликає течію струму у зворотному напрямі, і двигун обертатися у зворотному напрямі.

Регулювання швидкості досягнуте завдяки швидкому перемиканню двигунів. Драйвер двигунів перемкнеться між напрямом "вперед" і "стоп", примушуючи двигун обернутися вперед з пониженою швидкістю. Наприклад, якщо сигнал, що управляє, буде в "1" дві третини часу (робочий цикл 75 % часу), то двигун обертатиметься з потужністю 75 % від його максимальної швидкості.

Такий принцип управління швидкістю (серія пульсу змінної ширини) називають широко-імпульсною модуляцією (PWM). Як приклад PWM можна побачити на рисунку 1.7, де показано регулювання імпульсу від 100 % -го робочого циклу до 0 %, де частота обертання двигуна зменшується з максимальної швидкості до зупинки.

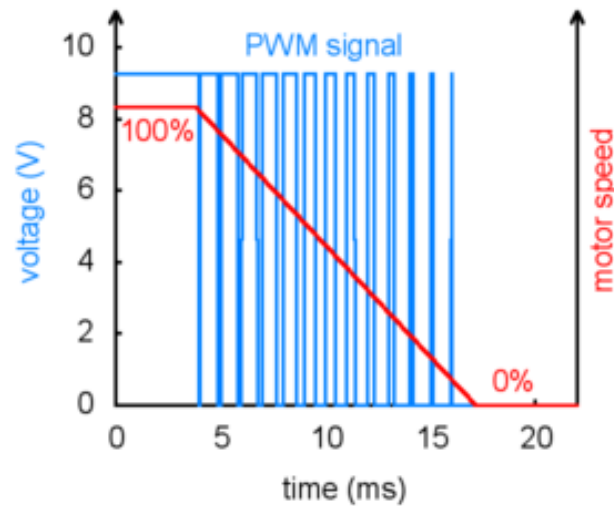


Рисунок 1.7 – PWM – управління швидкістю, що показує поступове уповільнення

Траєкторії руху роботу в залежності від співвідношення швидкостей обертання коліс показано на рисунку 1.8

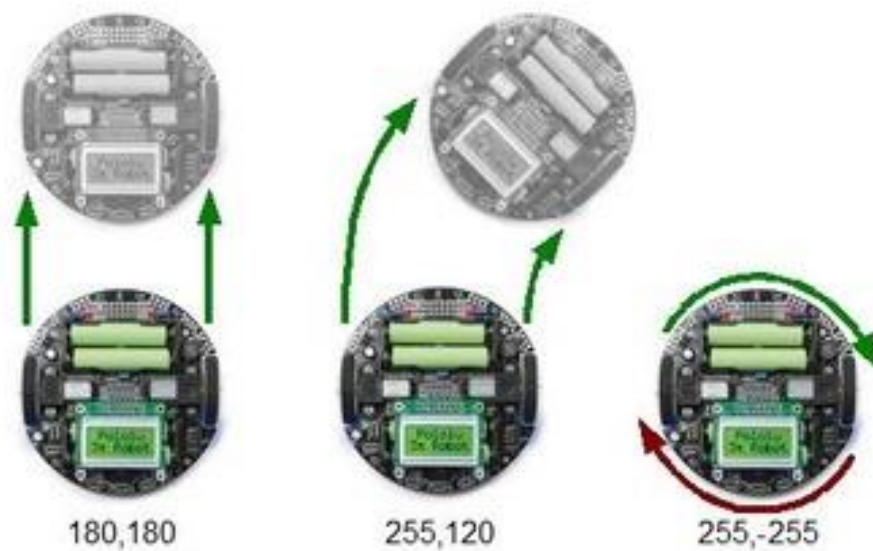


Рисунок 1.8 – Траєкторії руху роботу в залежності від співвідношення швидкостей обертання коліс

1.4 Аналіз структурної схеми мобільного робота

Структурна схема модуля керування мобільною платформою (МП) показана на рисунку 1.9.

Відеокамера отримує візуальні дані про оточуючу сцену та надсилає її до програми візуальної одометрії. За допомогою вищепри описаного алгоритму визначається траєкторія руху мобільної платформи. На виході програми формується потік даних [9].

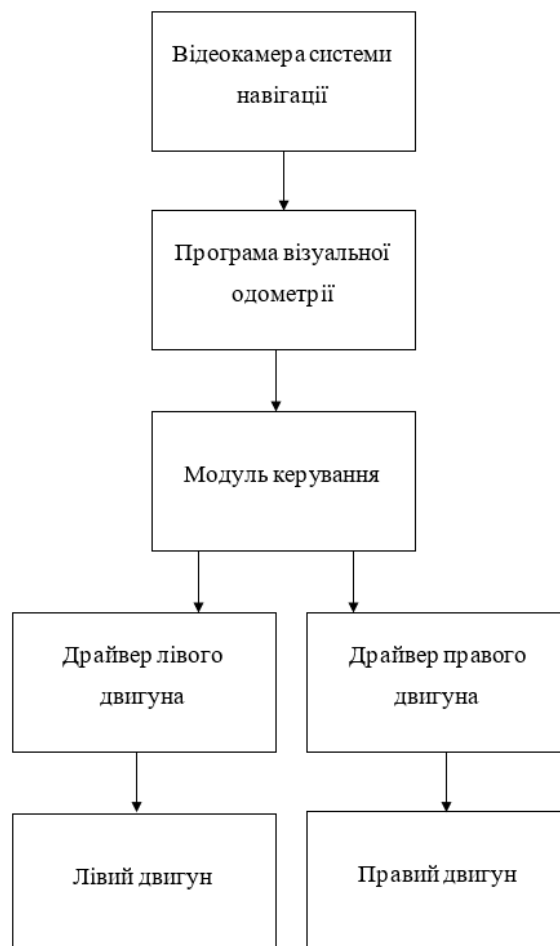


Рисунок 1.9 – Структурна схема мобільної платформи

На основі отриманих даних контролер визначає кут зміщення від траєкторії та обчислює коефіцієнт коригування швидкості двигунів приводу роботу.

Отримані коефіцієнти переводяться до числових значень частоти обертів двигунів. За допомогою драйверів, що підключені до контролері відбувається управління двигунами постійного струму.

Завданням драйвера руху є управління двома мотор-редукторами. В процесі руху платформи драйвер регулюватиме швидкість обертання мотор-редукторів, швидкість поворотів і кут поворотів. Він прийматиме команди з центрального процесора.

На рисунку 1.10 показана функціональна схема керування мобільної платформою.

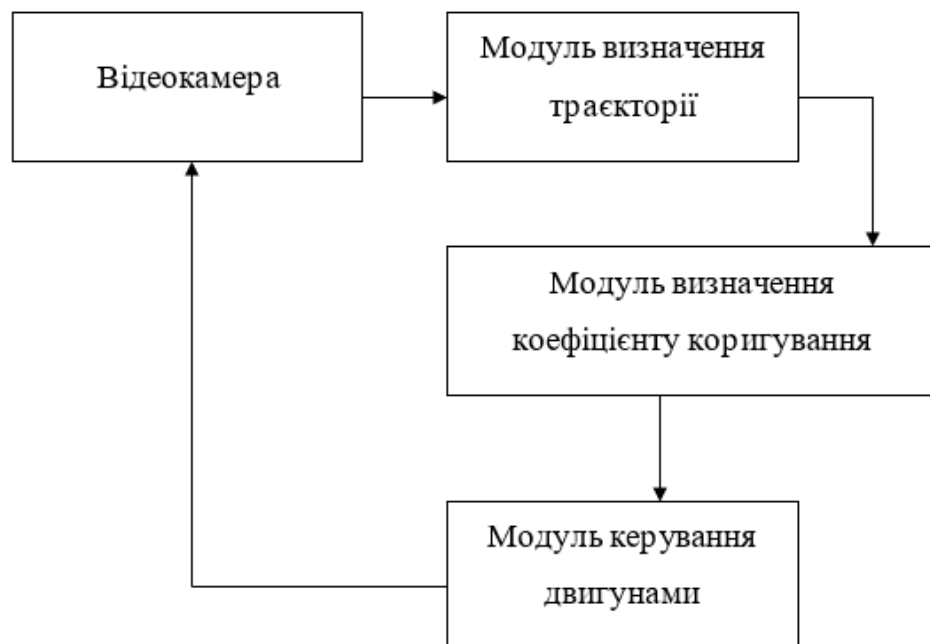


Рисунок 1.10 – Функціональна схема керування мобільної платформою

1.5 Постановка задач

Для досягнення поставленої мети необхідно вирішити наступні завдання:

- дослідити існуючі методи визначення траєкторії переміщення мобільного робота в виробничому середовищі серед перешкод;
- виконати аналіз існуючих принципів локалізації мобільних транспортних засобів в просторі та одночасного прокладення траєкторії руху;
- обрати програмну платформу для реалізації визначеного методу прокладення шляху;
- провести експериментальне дослідження сумісного використання методу локалізації та прокладення шляху в умовах невизначених перешкод;

1.6 Висновки по першому розділу

В результаті виконання першого розділу магістерської атестаційної роботи проведено аналіз існуючих методів навігації.

Розглянуто принцип визначення позиції і орієнтації робота шляхом аналізу послідовних зображень отриманих за допомогою камери – візуальна одометрія.

Описано загальний алгоритм візуальної одометрії та принцип отримання зображення з моно-камери.

Проведено аналіз принципу роботи мобільного робота, що використовується в роботі, наведено його зовнішній вигляд та схему роботи з двома диференційним приводом та двома колесами.

За результатами проведеного аналізу була розроблена функціональна схема керування мобільної платформи та структурна схема мобільної платформи.

2 МЕТОДИ ПРОКЛАДЕННЯ ШЛЯХУ В НЕВИЗНАЧЕНОМУ ПРОСТОРИ

2.1 Методи побудови карти простору

При використанні візуальної одометрії в якості сканера оточуючого простору ми отримуємо можливість побудови мапи навколишнього середовища поступово на кожному кроці переміщення мобільного роботу.

Технологія SLAM (одночасне побудова карти і визначення місця розташування) на сьогоднішній день є передовою в питанні навігації мобільних роботів [10]. Дана технологія необхідна для успішного переміщення робота по невизначеній місцевості і полягає в побудові карти навколишнього середовища й відстеження його розташування в цьому середовищі, що здійснюється його бортовим комп'ютером.

Як правило, інформацію про навколишній простір бортова система робота отримує за допомогою датчиків, таких як лазерні й ультразвукові далекоміри, одна, дві або відеокамери. Ультразвукові та лазерні датчики є досить дорогими і, що найголовніше, масивними пристроями, а використання відеокамер дає набагато більше інформації (колір, освітленість і т. д.), але забезпечує далеко не повний огляд навколишнього простору.

Для автономної навігації в системі керування мобільним роботом слід побудувати карту робочого середовища. Алгоритм SLAM може виконувати процеси картографування, як показано на рисунку 2.1.

Для досягнення мети побудови карти робочого середовища запропоновано багато методів SLAM. Перший підхід заснований на моніторингу положення робота в реальному часі, який розміщується в будь-якому місці навколишнього середовища під час процесу побудови карти. Найчастіше застосовуваний метод для цього підходу – використання Gmapping [11, 12]. Дані одометрії, які описують положення робота, можна отримати, використовуючи дані, що надаються поєднанням хорошого

датчика і хорошого обчислювального алгоритму, або шляхом злиття декількох датчиків, таких як IMU або GPS, що використовуються в навігації. У більшості випадків розташування робота в кадрі, отриманому в результаті зчитування з відеокамери з часом змінюється. Це пов'язано з дефектами обладнання та шумом сенсору відеокамери, тому не завжди рекомендується покладатися на один датчик для оцінки позиції роботи.

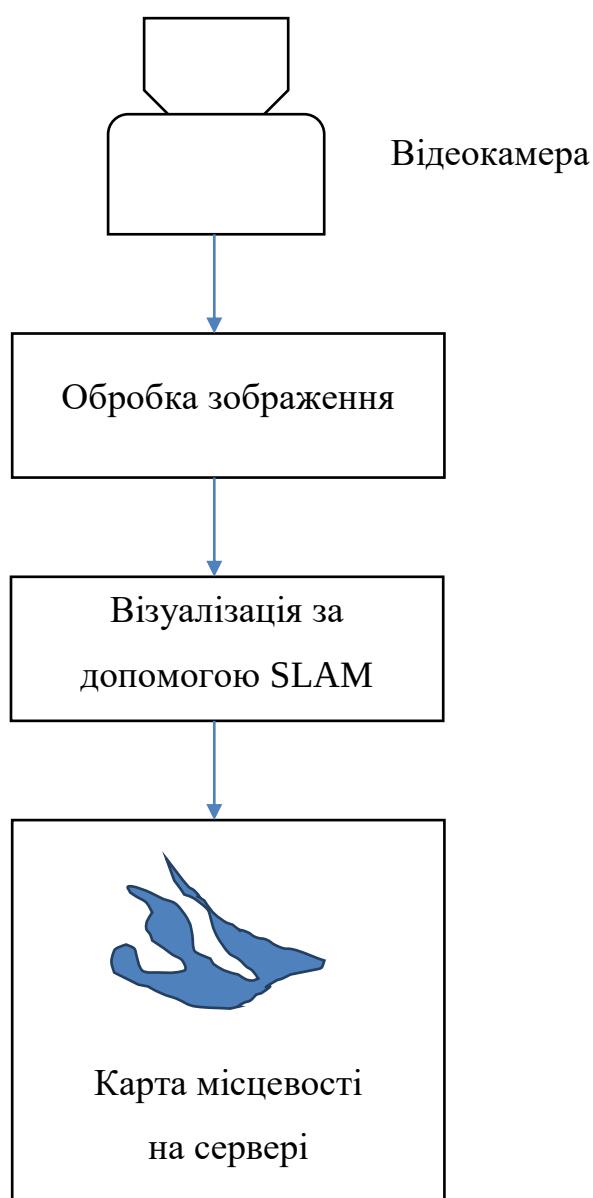


Рисунок 2.1 – Процеси картографування в реальному часі

Для вирішення проблеми Алгоритм фільтрації, такий як адаптивна локалізація Монте-Карло (AMCL) [13], може вдосконалити інформацію про одометрію та підтримувати взаємозв'язок між координатами глобальної карти та локальної. AMCL – це техніка, яка використовує фільтр частинок у режимі реального часу для фільтрації шуму в одометрії для оцінки більш точного положення робота в навколишньому середовищі.

На багатьох робочих платформах інформацію про одометрію часто важко точно розрахувати, якщо вона покладається на дані, які надає енкодер колеса через проблеми зі пробуксовкою колеса. Щоб подолати цю проблему, використовується інший підхід, який використовує високошвидкісний діапазон датчиків зору для оцінки та підтримання позиції робота шляхом узгодження даних про позицію, що отримані з датчиків, коли робот рухається по невідомій області. Інформацію про переміщення, обертання колес та швидкість можна отримати за допомогою методів виявлення та узгодження ознак одометрії.

Цей підхід дуже схожий на побудову панорамного виду, де кілька частково перекриваються зображень вигляду збираються, щоб отримати зображення великого поля зору. Однією з переваг цього методу є те, що інформацію про місцезнаходження робота в реальному часі можна оцінити на основі відповідних особливостей даних візуального датчика, навіть не залежно від енкодерів коліс або магнітних датчиків, що часто показують помилкові дані внаслідок ковзання колеса або перешкод у зовнішньому магнітному полі. Недоліком цього підходу є те, що він вимагає якісного датчика та складних процесів у реальному часі для виявлення подібності між кадрами.

Нещодавно стали популярні лазерні датчики з високою швидкістю сканування в широкому полі зору (LIDAR) та надійними методами узгодження функцій. Такі пристрої наряду з засобами візуальної одометрії роблять цей підхід більш простим та ефективним у оцінці позиції роботів в просторі.

Після побудови карти робочого середовища за допомогою векторного відображення, карти зберігаються на сервері карт, щоб можна було в подальшому використати можливість автономної навігації.

Збережена карта включає два файли: файл зображення у форматі Рunt та файл конфігурації у форматі XML. Файл зображення визначає значення пікселів, що відповідають конкретним елементам у реальному середовищі. Кожен піксель визначається на основі роздільної здатності карти в системі навігації. Роздільна здатність пікселів дорівнює роздільній здатності сітки в цьому документі. Зокрема, для кожного розташування пікселя (x, y) на карті попередньої збірки значення пікселя $M(x, y) = 0$ відповідає пікселю вільного простору, $M(x, y) = 100$ являє собою піксель перешкоди або межа кімнатного пікселя, а $M(x, y) = -1$ відповідає невизначеному пікселю, де об'єкт неможливо відсканувати.

Інформація у файлі конфігурації описує положення початку (X_0, Y_0) , масштаб карти M_s , ширину карти M_w та висоту карти M_h . Вісь x та вісь y рухаються відповідно по ширині та висоті карти. На основі інформації у файлі конфігурації рівняння (2.1) та (2.2) використовуються для відповідності між розташуванням пікселів (x, y) у файлі зображення та його координатою (C_x, C_y) у реальному середовищі:

$$C_x = X_0 + (x - M_w/2)M_s, \quad (2.1)$$

$$C_y = Y_0 + (y - M_h/2)M_s. \quad (2.2)$$

Попередньо побудована карта поділяється на сітку із заздалегідь визначеним розміром квадратів, наприклад, $0,1$ м. Одна квадратна одиниця складається з її розташування на (x, y) та її довжини. Перетворення піксельних координат у фактичні координати положення в метрах буде здійснено за допомогою рівнянь (1) та (2). Відстань між двома сусідніми комірками – це мінімальна відстань кроку робота, що рухається в просторі.

2.2 Методи прокладення шляху в просторі на основі алгоритму A*

Кінцевою метою вирішення проблем покриття території є охоплення вільних просторових площ, визначених на попередньо побудованій карті, шляхом автоматичної навігації та слідування заздалегідь визначеному шляху, створеному глобальним планувальником маршруту. З цією метою запропоновано декілька методів прокладення шляху в просторі.

Звичайні офлайн-методи вибирають точки переміщення вручну, знаючи форми та розміри карт, а також форми та положення перешкод для створення шляху покриття області. Крім того, промислові мобільні роботи використовують датчики зіткнення для виявлення та дотримання меж перешкод при формування зони покриття.

Алгоритм "зірка" [14] – найпопулярніший і широко використовуваний метод для обчислення та планування шляху для роботи автономної навігації, уникаючи перешкод. Основна ідея цього алгоритму полягає у пошуку найкоротшого шляху між початковою точкою та точкою призначення на основі функції витрат.

Алгоритм "зірка" базується на сітковій карті. Клітини сітки визначаються як: вихідна клітина, клітина призначення, клітини вільного простору та комірки перешкод.

Рисунок 2.2, а описує один із таких прикладів алгоритму пошуку найкоротшого шляху від комірки 14 до комірки 35. На карті сітки є вісім сусідніх комірок навколо комірки, як показано на рисунку 2.2, б. На першому кроці алгоритму «зірка» кожній сусідній комірці вільного простору навколо вихідних комірок присвоюється відповідна вартість, така як горизонтальна та вертикальна. Сусідні комірки мають вартість 10, тоді як діагональні сусідні комірки мають вартість 14. Сусідні комірки перешкоди не мають ніяких витрат. Тоді вартість цих вільних просторових сусідів накопичуватиметься, поки не буде досягнута комірка призначення.

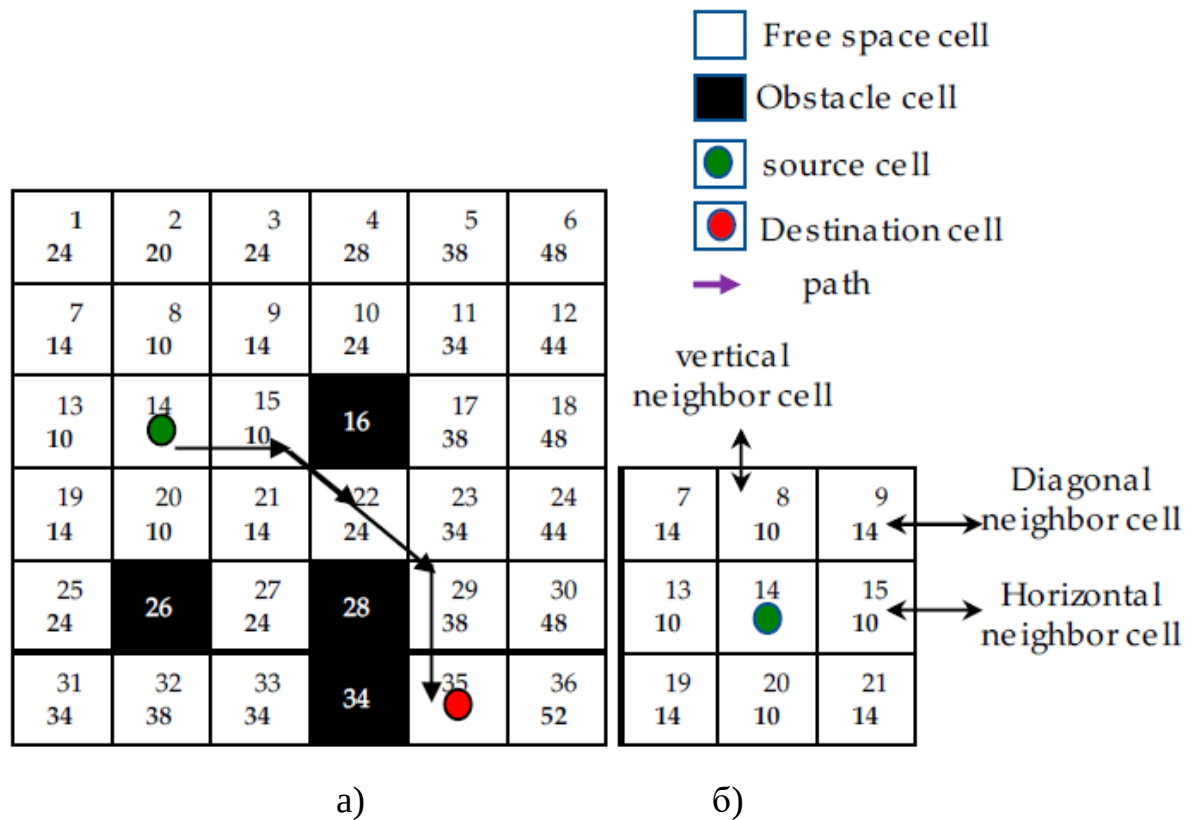


Рисунок 2.2 – Принцип визначення траєкторії за алгоритмом A*

Найкоротший шлях вибирається шляхом пошуку назад, щоб знайти комірки з найменшим значенням від комірки призначення до комірки джерела. У разі максимізації покриття площі з можливістю уникнути перешкод, може бути використаний підхід A-зірки, що базується на зигзагоподібному вимірюванні: визначивши набір розташування шляхових точок на межі, як на зигзагоподібному візерунку, а потім дозволивши алгоритму «зірка» обчислити найкоротший шлях до очищення цих шляхових точок. Для охоплення всієї площі, визначеної межею на карті, траєкторія являє собою зигзагоподібну лінію, яка складається з рівномірно розташованих сегментів. Кінцева точка кожного сегмента з'єднана з початковою точкою наступного сегмента.

Звичайний алгоритм A-зірка має обмеження, що не охоплює клітини вільного простору у визначеному діапазоні та переглядає клітини вільного

простору, які вже були охоплені. Якщо припустити, що ширина робота дорівнює довжині 2-х комірок, недоліки алгоритму А-зірка можна побачити на рисунку 2.3.

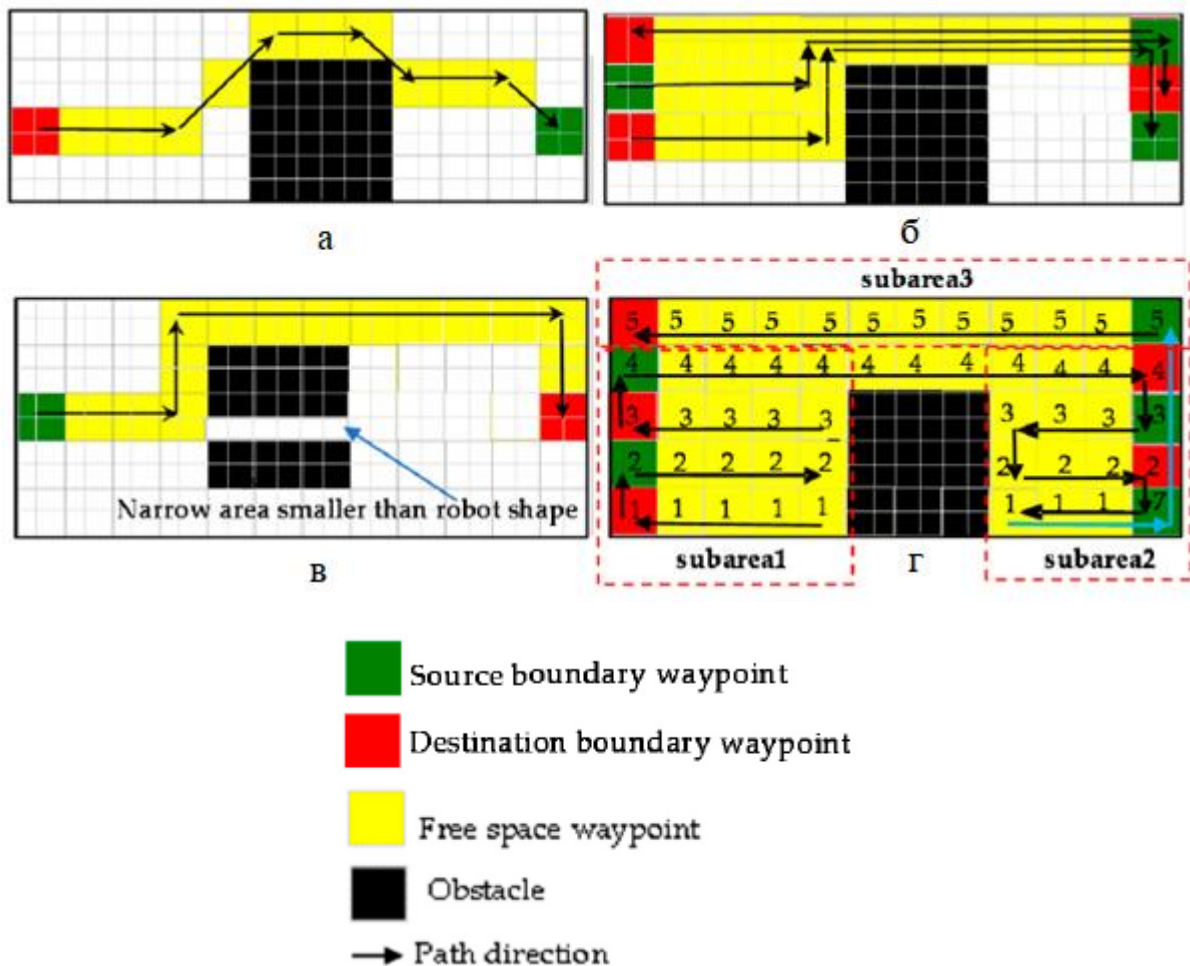


Рисунок 2.3 – Недоліки традиційних методів алгоритму А*

На рисунку 2.3, а (алгоритму А* з діагональним рухом) можна бачити, що коли спрацьовує діагональна особливість прокладення короткої траєкторії руху від джерела до пункту призначення за алгоритмом А*, деякі комірки навколо діагональної траєкторії не будуть покриті.

На рисунку 2.3, б (алгоритму А*, що охоплює зону) наведено випадок, коли на шляху з'являється перешкода, що з'єднує дві шляхові точки, то є два варіанти вибрати шлях, щоб уникнути перешкод, піднімаючись або спускаючись вниз. Якщо алгоритм А* вирішить рухатись, досягнувши

комірок перешкоди, як на рисунку 2.3, б, деякі комірки під перешкодою залишаються непокритими, а покриті комірки над перешкодами будуть переглянуті, щоб знайти шлях, що з'єднується праворуч з лівою граничною точкою. Точно так само верхні клітини не будуть покриті, а нижчі клітини будуть переглянуті, якщо A^* вирішить спуститися, дійшовши до клітин-перешкод. Більше того, алгоритм A^* вибирає шлях, який проходить навколо вільного простору, визначеного як пікселі вузького простору, як показано на рисунку 2.3, в (відкритий вузький простір методом фіксованої морфології).

Модифікований метод зигзагоподібного сканування на основі алгоритму A^* для проблеми покриття території був запропонований у дослідженні [15]. У цьому методі карта поділяється на підкарти, що межують з межею перешкоди. Тоді ці підрайони охоплюються зигзагоподібним плануванням. Після закінчення кожної підобласті алгоритм вибирає наступну непокриту область, даючи найкоротшу відстань із поточною охопленою. Далі алгоритм A^* буде використаний для пошуку шляху до цієї непокритої області, а цей шлях називається шляхом виходу, як на рисунку 2.3, г (вкладені простори алгоритму A^* , що базується на покритті площини).

Однак у цього алгоритму є деякі обмеження. По-перше, розглядається алгоритм перешкод із відносно простими формами. По-друге, при переході до наступної непокритої підкарти, якщо остання покрита комірка підкарти розташована в місці, де немає сусідніх комірок із наступної непокритої підкарти, деякі комірки поточного підрайона потрібно переглянути, щоб дістатися до непокритого підрайона (на рисунку 2.3, г шляхи представлені синіми стрілками). Також, проблема покриття вузьких просторових обмежень цим методом не вирішена. Ця проблема неминуча для фіксованих морфологічних робіт, які не можуть змінити свою морфологію.

У цій роботі пропонується використовувати модифіковане зигзагоподібне сканування на основі алгоритму A^* для планування траєкторії, включаючи набір проміжних точок для навчального роботу NTech-Robot.

Зокрема, форми та розміри карт, отримані на етапі візуальної одометрії алгоритмом SLAM, можуть бути встановлені випадковим чином або отримані динамічно в режимі реального часу.

Отже, зони, які потрібно покрити, та характеристики перешкод, такі як форми та розташування, а також межі карти не відомі перед створенням шляхів покриття.

Запропонований метод використовує особливості карт для визначення різних типів шляхових точок, включаючи граничні шляхові точки, шляхові точки перешкод автономно. Крім того, на відміну від традиційного планування шляху, що відповідає лише розташуванню пунктів маршрутних точок, шляхові точки в запропонованому методі включаються як точки розташування динамічних перешкод визначені методом візуальної одометрії.

Таким чином, можна охопити максимальну площу місцевості, та мінімізувати проблему перегляду вже розглянутих територій. Також можна досягти результату коли робот буде орієнтуватися через вузькі просторові обмеження карт.

Процес створення зигзагоподібного планування шляху для карти зображено на рисунку 2.4.

Траєкторія, зображена на рисунку 2.4, встановлюється для початку з першої точки шляху, а потім йде за жовтим шляхом до останньої точки. Перша точка шляху визначається як точка початку, розташована в нижньому лівому куті. Необхідно звернути увагу, вісь x позначається зеленим кольором, а вісь y – червоним.

Оскільки перешкоди, розташовані випадково на карті і мають довільну форму, а карта сегментована на сітки, алгоритм A^* використовується для пошуку найкоротшого шляху, щоб уникнути перешкод, і з'єднати кожну пару автоматично згенерованих шляхових точок.

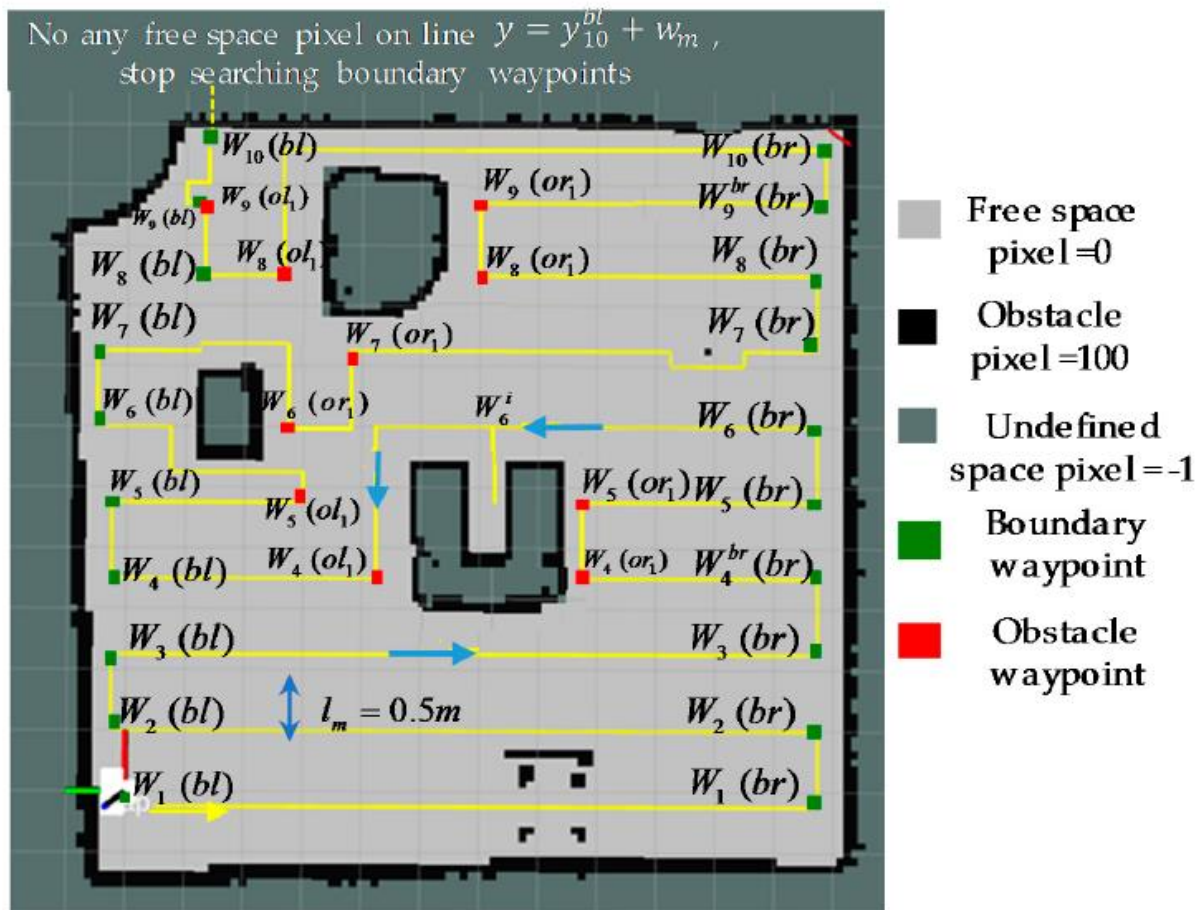


Рисунок 2.4 – Створення зигзагоподібного планування шляху для карти

На даному рисунку для даної карти рух у діагональному напрямку в алгоритмі A* вимкнено для забезпечення максимального покриття площі.

2.3 Висновки по другому розділу

В результаті виконання другого розділу магістерської атестаційної роботи проаналізовані методи прокладення шляху в невизначеному просторі. Розглянута технологія SLAM – одночасне побудова карти і визначення місця розташування. В результаті проведеного аналізу запропоновано використовувати для побудови шляху на карті виробничого простору алгоритму A*.

Запропонований метод використовує особливості карт для визначення різних типів шляхових точок, включаючи граничні шляхові точки, шляхові точки перешкод автономно. Крім того, на відміну від традиційного планування шляху, що відповідає лише розташуванню пунктів маршрутних точок, шляхові точки в запропонованому методі включаються як точки розташування динамічних перешкод визначені методом візуальної одометрії.

3 ДОСЛІДНО-АНАЛІТИЧНИЙ РОЗДІЛ

3.1 Опис запропонованого методу визначення траєкторії руху в невизначеному просторі

Для реалізації поставленої мети та рішення задачі прокладення оптимального шляху переміщення серед невизначених перешкод пропонується наступний метод.

Суть запропонованого методу полягає в поєднанні методу пошуку оптимальної траєкторії руху, коли всі перешкоди визначені, та методу візуальної одометрії для визначення змін в навколишньому просторі.

Поєднавши ці два методи ми отримуємо можливість постійного пошуку та аналізу розташування нових перешкод, що з'являються в оточуючому просторі, та швидкого реагування на знайдені зміни в реальному часі.

Перед адаптивною візуальною системою ставляться такі завдання:

- визначення місця розташування робота в початковій позиції і при переміщенні:
- орієнтація робота в просторі і відносно початкової точки:
- визначення і стеження за об'єктами, що знаходяться в робочому просторі:
- виділення непрохідних ділянок.

Для аналізу навколишнього середовища буде застосовуватись метод візуальної одометрії, суть якого викладена в попередніх розділах. Даний метод дає змогу прив'язати перешкоди до локальної мапи місцевості. Знайдені зміни відразу наносяться на мапу та в подальшому використовуються для прокладення оптимального маршруту [17].

Прокладення маршруту за даними карти відбувається з використанням алгоритму A^* , що був представлений в попередньому розділі.

Таким чином ми маємо три незалежних процеси: пошук перешкод, модернізація мапи та прокладення маршруту. Запропонований метод схожий з відомим принципом переміщення роботів в просторі, який має назву SLAM (метод одночасної навігації і побудови мапи).

Відмінність полягає в тому, що в традиційному методі робот повністю не має уяви про оточуючий простір та першим кроком проводить дослідження простору та побудову мапи.

В нашому методі мапа вже відома і завантажена до пам'яті мобільного пристрою. Наприклад, це дуже часто буває, коли робот працює на виробничій ділянці, де заздалегідь відомо місце розташування обладнання.

Тому, на першому етапі відбувається швидке прокладення шляху від початкової точки до заданої цілі. В процесі переміщення роботу він постійно сканує простір та стежить за змінами в оточуючому просторі. Зміни можуть бути викликані появою людини перед мобільним пристроєм, появою іншого мобільного робота, або переміщенням продукції.

Таким чином, визначивши місце нової перешкоди або її зникнення, модуль керування роботом перебудовує траєкторію руху, та визначає оптимальний маршрут для даної поточної ситуації.

Таким чином, запропонований алгоритм покладений у наступну модель адаптивної системи керування мобільною платформою в невизначеному середовищі, що зображена на рисунку 3.1.

Навколишнє середовище складається з об'єктів, що не рухаються та є елементами обладнання (O) або конструкцій приміщення, а також рухомих об'єктів, таких, як людина, або інші мобільні роботи (U).

Таким чином, кожна точка простору може бути представлена $P^* \{[O1, O2...], [U1, U2...]\}$.

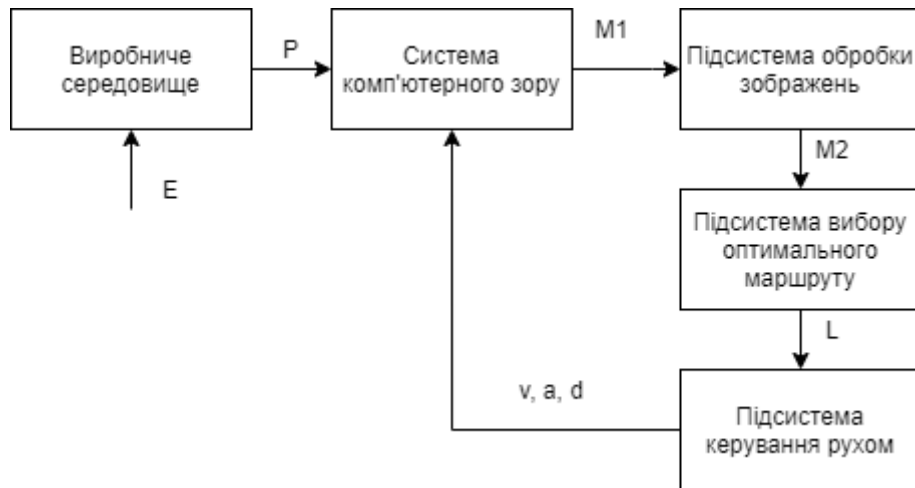


Рисунок 3.1 – Модель адаптивної системи керування мобільною платформою

Навколишнє середовище можна описати набором опорних точок, що складають просторовий фон який заважає, або не заважає руху:

$$P = \{P^*1, P^*2, \dots P^*n\},$$

де P^* – координати опорної точок об'єктів приміщенні;
 n – кількість опорних точок в простору.

$$M1 = \{S, W\},$$

де S – метод визначення візуальних координат;
 W – параметри методу.
 На

$$L = \{\Delta X, \Delta d\},$$

де X – локалізація мобільної платформи;
 d – напрям руху платформи.

На виході підсистеми керування рухом отримуємо параметри впливу на рух мобільної платформи, а от же і на зміщення системи комп'ютерного зору, що є жорстко пов'язана з платформою (v , a , d). Параметр v описує зміну швидкості руху платформи. Параметр a описує кутове зміщення, а параметр d – напрям руху.

3.2 Вибір методу візуальної одометрії

В результаті аналізу існуючих методів реалізації візуальної одометрії порівняні два найпоширеніших методи: ORB-SLAM2 та LSDSLAM.

ORB-SLAM2 – це бібліотека SLAM у реальному часі для монокулярних, стерео та RGB-D камер, яка обчислює траєкторію камери та розріджену 3D-реконструкцію (у випадку стерео та RGB-D із справжнім масштабом). Він здатний виявляти петлі та переміщувати камеру в режимі реального часу [18].

Даний метод заснований на детекторі ключових точок [20]. Висока швидкість детектора відеокамери дозволяє методу працювати в реальному часі в умовах обмежених обчислювальних ресурсів.

На рисунку 3.2 показана блок-схема, що пояснює принцип роботи алгоритму. Його робота розділена на три основні потоки:

- відстеження кадрів (Motion Estimation);
- побудова карти (Local Mapping);
- замикання кадрів (Loop Closing).

Як можна бачити з наведеного рисунку, потік Motion Estimation приблизно визначає поточний стан камери шляхом пошуку схожого кадру в локальній карті і зіставлення ключових точок з знайденим кадром.

Потік Local Mapping виконує побудову карти поблизу поточного положення камери і оптимізує карту.

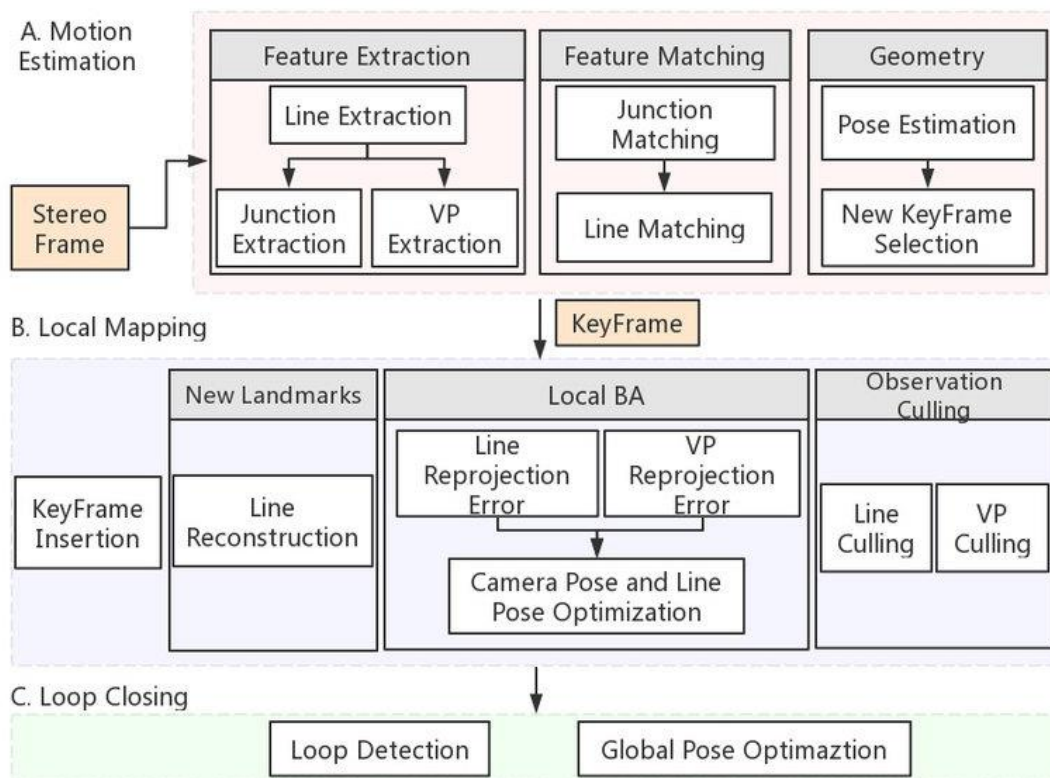


Рисунок 3.2 – Принцип роботи ORB-SLAM2 алгоритму

Loop Closing – алгоритм замикання циклів, який шукає і об'єднує схожі кадри.

На першому кроці алгоритму виконується ініціалізація карти, яка складається з набору точок (MapPoints) і ключових кадрів (KeyFrames). Ключові кадри зберігають інформацію про стан камери в ключових точках, присутніх в отриманому кадрі.

Далі, з використанням карти точок і ключових кадрів виконується побудова неорієнтованого зваженого графа перетинів. В даному графі кожен кадр є вузол. Пара вузлів зв'язується ребрами, якщо у відповідній парі кадрів спостерігається більше 15-ти спільних точок карти. Вагою ребер є число спільних точок.

Потік Tracking відстежує переміщення камери. Він витягує ключові точки за допомогою даного алгоритму, а потім намагається зіставити їх з попереднім кадром. У разі невдачі виконується релокалізація.

У разі успішного зіставлення в графі перетинів ключових кадрів шукається локальна карта. Після цього проводиться зіставлення точок поточного кадру точки в локальній карті шляхом проекції. Далі виконується оптимізація локальної карти, за допомогою якої уточняється положення камери.

Модуль локального побудови карти обробляє нові кадри і додає їх в граф перетинів і основу графа. Крім того, він виконує локальну оптимізацію карти (Local BA) для отримання більш точної реконструкції хмари точок поблизу положення камери.

Модуль замикання циклів (LoopClosing) шукає схожі кадри для кожного нового кадру. Якщо такі кадри знайдені, то для них і поточного кадру обчислюється перетворення подібності.

Потім положення знайденого і поточного кадрів вирівнюються шляхом застосування знайденого перетворення, а однакові ключові точки об'єднуються.

Крім локальної оптимізації, ORB SLAM виконує глобальну оптимізацію карти, яка дозволяє зменшити накопичену помилку з урахуванням знайдених замикань циклів.

Інший метод визначення траєкторії – LSDSLAM. Даний алгоритм включає в себе три основних модуля [21]:

- Tracking;
- Depth Estimation;
- Map Optimization.

Блок-схема алгоритму LSDSLAM показана на рисунку 3.3.

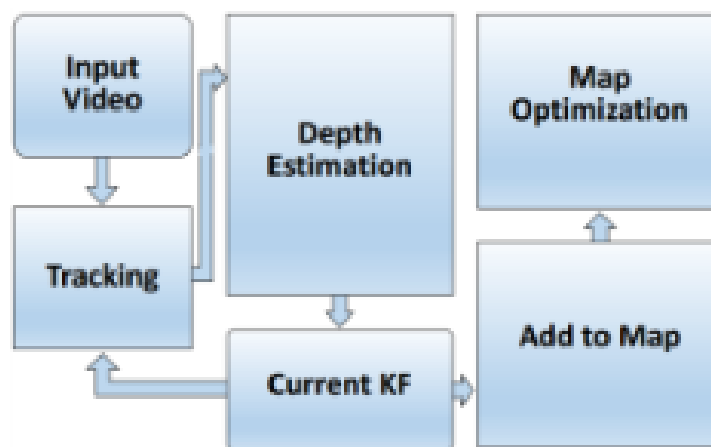


Рисунок 3.3 – Блок-схема алгоритму LSDSLAM

Модуль **Tracking** безперервно відстежує нові зображення з камери і визначає переміщення камери. Для оцінки переміщення обчислюється перетворення подібності між попереднім і новим кадрами. На відміну від ORB SLAM2, заснованому на зіставленні ключових точок, в LSD SLAM для розрахунку перетворення використовується мінімізація фотометричної помилки.

Модуль **Depth Estimation** порівнює новий кадр з поточним, а далі корегує або повністю замінює поточний кадр. Для порівняння використовується зважена сума відносної відстані від нового кадру до поточного і кутів повороту між ними. Якщо обчислена сума більше заданого порогу – поточний кадр замінюється новим.

Модуль **Map Optimization** виконує оптимізацію карти. Оптимізація дозволяє запобігти накопиченню помилок у відстеженні місця розташування і підтримує точність побудови карти навколишнього середовища. Оптимізація виконується окремою бібліотекою безперервно в окремому потоці.

Для зберігання карти навколишнього середовища використовується граф. Кожен вузол графа зберігає зображення і, відповідну йому, зворотну карту глибини. Вузли з'єднуються ребрами, які містять знайдене перетворення подібності між двома зображеннями.

При порівнянні алгоритмі використані відкриті ресурси та бібліотеки на ресурсі GitHub. Буди отримані наступні результати:

- обидва алгоритми мають відкритий вихідний код, і опубліковані під ліцензією GPL;
- обидва алгоритми мають інтеграцію з середовищем ROS (Robot Operating System), але алгоритм LSD SLAM не підтримується в останніх версій ROS;
- алгоритм LSD SLAM має реалізацію тільки для монокулярної камери в той час ORB SLAM підтримує роботу з монокулярною, стерео і RGB-D камерами;
- алгоритм ORB SLAM значно менш вимогливий до ресурсів, і здатний працювати в реальному часі, наприклад, на мікрокомп'ютері Raspberry PI. В той же час алгоритм LSD SLAM менш продуктивний, оскільки зберігає і обробляє істотно більший обсяг інформації (зображення цілком, а не тільки ключові точки), і для прийнятної роботи вимагає значних обчислювальних ресурсів;
- на відміну від алгоритму ORB SLAM, LSD SLAM здатний будувати щільну карту навколишнього середовища, яку можна використовувати для планування шляху в середовищі з перешкодами.

Таким чином, в результаті порівняння можемо зробити висновок, що алгоритм ORB SLAM2 найбільше підходить для подальшого використання та може бути використаний для моделювання поведінки мобільної платформи в просторі.

3.3 Розробка алгоритму вирішення задачі пошуку шляху переміщення

Використовуючи попередні висновки та обраний алгоритм візуальної одометрії побудуємо алгоритм вирішення задачі пошуку шляху переміщення в виробничому середовищі серед невизначених перешкод, який представлено на рисунку 3.4.

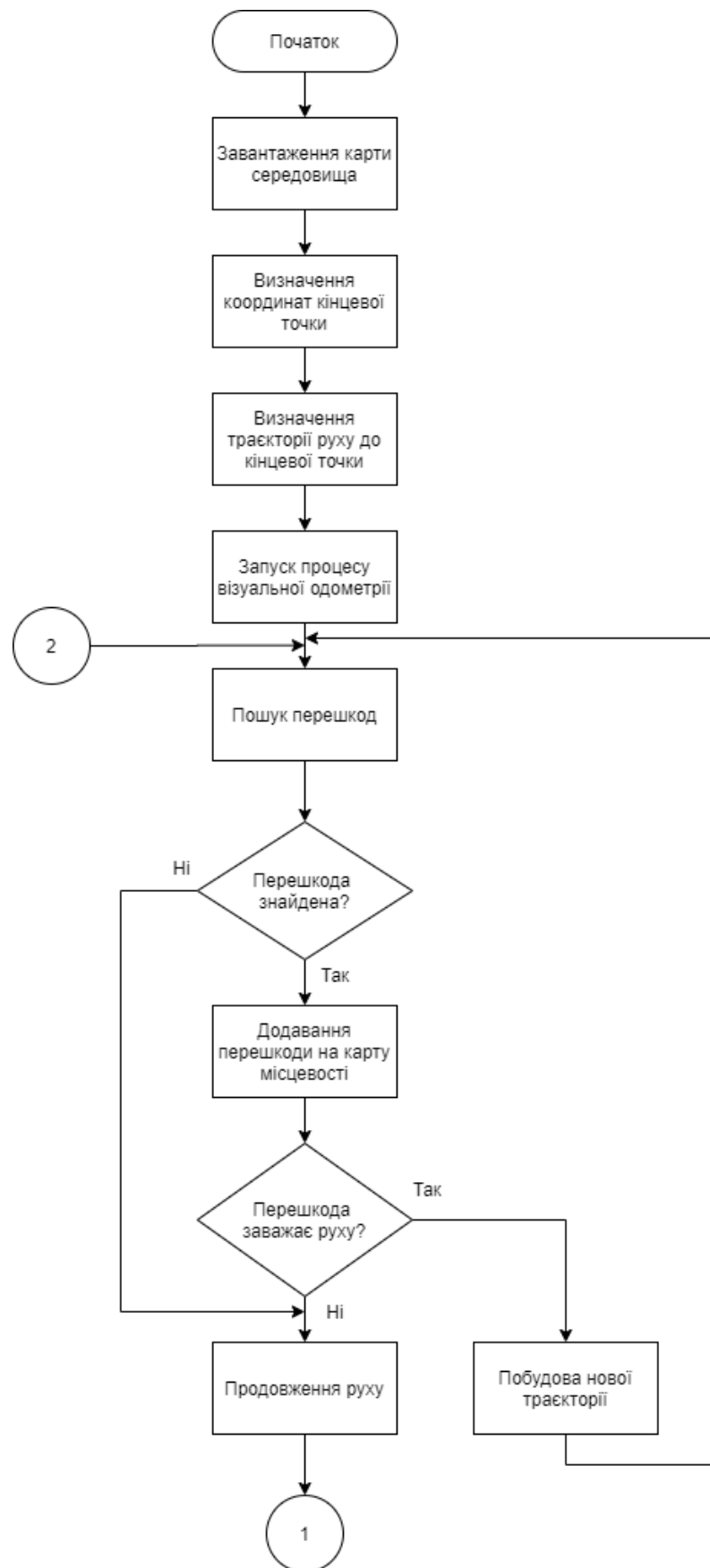


Рисунок 3.4 – Алгоритм вирішення задачі пошуку шляху переміщення в виробничому середовищі серед невизначених перешкод

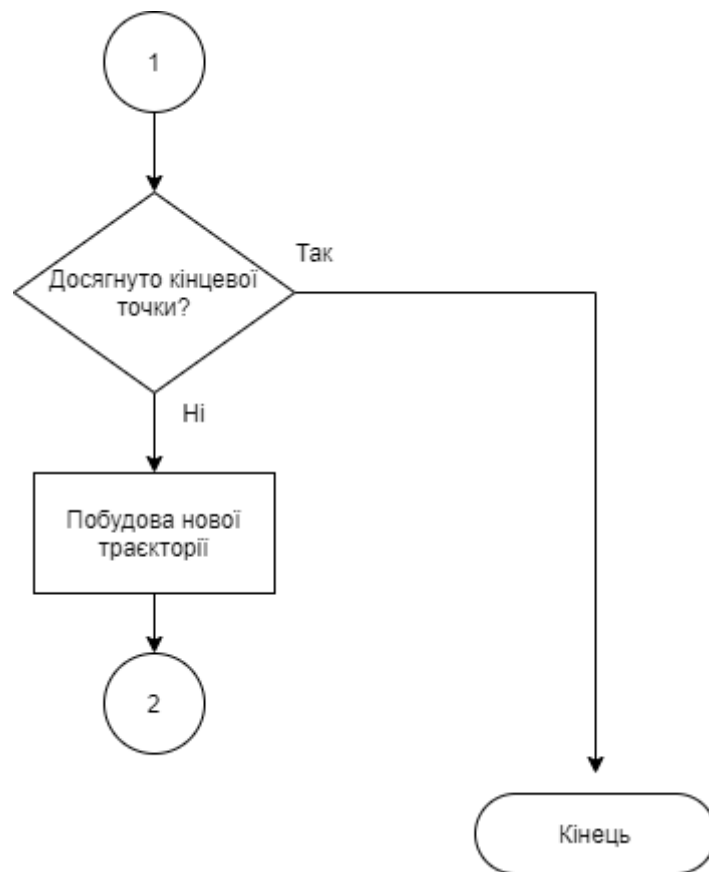


Рисунок 3.4, аркуш 2

Для побудови маршруту руху використовується глобальна інформація про середовище в якому відбувається переміщення, наприклад, у вигляді матриці, кожен елемент якої відповідає певній ділянці поверхні.

Індекси окремого елемента матриці визначають лінійні координати ділянки місцевості, а значення елемента – стан місцевості.

Управління локальними переміщеннями відомим маршрутом здійснюється на підставі інформації про характер місцевості в ближній зоні зору робота.

Якщо визначити маршрут руху як послідовність опорних пунктів руху, що включає початкове і кінцеве положення робота, то завдання побудови оптимального маршруту включає формування деякого набору траєкторій і подальший вибір такої підмножини точок, яке оптимізує рух робота.

Для цього використовуються різні алгоритми пошуку траєкторії руху.

Процесу прокладки маршруту руху робота передуює складання карти середовища. Якщо при визначенні перешкоди та нанесенні її на карту буде визначено, що вона не заважає руху, то мобільна платформа продовжить їхати за шляхом, що раніше було прокладено.

Якщо перешкода заважає руху, то траєкторія буде скорегована і обрана та, що є оптимальною для поточного стану мобільної платформи.

3.4 Висновки по третьому розділу роботи

В результаті виконання третього розділу магістерської атестаційної роботи розглянуто обраний метод визначення траєкторії руху в невизначеному просторі. Суть запропонованого методу полягає в поєднанні методу пошуку оптимальної траєкторії руху, коли всі перешкоди визначені, та методу візуальної одометрії для визначення змін в навколишньому просторі. Поєднавши ці два методи ми отримуємо можливість постійного пошуку та аналізу розташування нових перешкод, що з'являються в оточуючому просторі, та швидкого реагування на знайдені зміни в реальному часі.

Для аналізу навколишнього середовища буде застосовуватись метод візуальної одометрії, суть якого викладена в попередніх розділах. Знайдені зміни відразу наносяться на мапу та в подальшому використовуються для прокладення оптимального маршруту. Прокладення маршруту за даними карти відбувається з використанням алгоритму A^* .

Побудована модель адаптивної системи керування мобільною платформою. Розроблено алгоритм вирішення задачі пошуку шляху переміщення.

4 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ

4.1 Опис методики проведення експериментальних досліджень

Для перевірки працездатності запропонованого методу прокладення оптимального шляху переміщення серед невизначених перешкод будемо використовувати імітаційне (програмне) моделювання, при якому логіко-математична модель досліджуваного об'єкта є алгоритм функціонування об'єкта, реалізований у вигляді програмного засобу для комп'ютера.

Призначення комп'ютерного моделювання полягає в отриманні кількісних і якісних результатів запропонованої моделі. Якісні результати аналізу виявляють невідомі раніше властивості системи, що моделюється: її структуру, динаміку розвитку, стійкість, цілісність та ін. Кількісні висновки в основному носять характер аналізу існуючої системи або прогнозу майбутніх значень деяких змінних.

Існують різні види імітаційного моделювання [26]:

- статистичне – дозволяє прогнозувати розвиток явища, та зрозуміти напрямки його розвитку;
- агентне відрізняється від інших тим, що в ньому задаються, характеристики окремих її об'єктів (агентів), а загальна характеристика роботи системи і загальна динаміка складаються з накладення друг на друга характеристик агентів;
- моделювання динамічних систем – це поведінка системи в часі в просторі;
- системна динаміка – моделі потокового типу, безперервне моделювання;
- дискретно-подієве – передбачає вирішення завдань щодо роботи системи, що моделюється;

– ситуаційне – це пошук найбільш оптимального плану дій в умовах виникнення непередбачених ситуацій;

– когнетивне – аналогічний методу системної динаміки, тільки в ньому закладаються специфічні об'єкти, в неї додаються елементи які розраховуються на основі знань, уявлень.

Виходячи з особливості процесу, що моделюється, ми обираємо ситуаційне моделювання. Прокладення оптимального шляху повинне моделюватися в умовах наближених до реального застосування мобільних платформ. Треба забезпечити можливість впливати на хід експерименту, наприклад, шляхом додавання нових перешкод вже після початку роботи алгоритму (програми). Це може робити оператор натискаючи на віртуальну карти виробничого середовища в будь-якому місці і тим самим додавая новий об'єкт, який треба враховувати при розрахунках.

Для того, щоб максимально точно відтворити поведінку реального мобільного пристрою в віртуальному середовищі необхідно розробити незалежні програмні процеси. Ці процеси будуть працювати паралельно та взаємодіяти лише на рівні подій. Такими процесами будуть:

- сканування простору та пошук перешкод;
- побудова траєкторії руху на основі даних про знайдені перешкоди;
- переміщення цифрового двійника на віртуальній карті виробничого середовища.

Для взаємодії між процесами будуть створені загальні ділянки пам'яті до яких можуть звертатись програмні функції. Наприклад, процес пошуку перешкод буде зберігати знайдені координати в динамічному масиві. До цього масиву також має доступ процес прокладення шляху. Результат роботи алгоритму пошуку оптимального маршруту буде збережено в іншому масиві, до якого матиме доступ процес переміщення мобільної платформи в просторі.

Таким чином ми реалізуємо задачу незалежного існування декількох віртуальних процесів. Для синхронізації дії всіх процесів треба передбачити загальну змінну, яка буде пов'язана з загальним (модельним) часом.

Існують два основних способи зміни модельного часу:

- покроковий (застосовуються фіксовані інтервали зміни модельного часу);
- подієвий (застосовуються змінні інтервали зміни модельного часу, при цьому величина кроку вимірюється інтервалом до наступної події).

У разі покрокового методу просування часу відбувається з мінімально можливою постійною довжиною кроку. Ці алгоритми не надто ефективні з точки зору використання машинного часу на їх реалізацію.

При використанні подієвого методу (принцип "особливих станів") значення модельного часу змінюється тільки коли змінюється стан системи. В подієвих методах довжина кроку тимчасового зсуву максимально можлива. Модельний час з поточного моменту змінюється до найближчого моменту настання наступної події. Застосування подієвого методу переважно в разі, якщо частота настання подій невелика, тоді довжина кроку дозволить прискорити хід модельного часу.

Для виконання імітаційного моделювання будемо використовувати інтегроване середовище VisualStudio та мову програмування C#.

4.2 Опис структури програми для проведення моделювання

Виходячи з попереднього аналізу, програмний засіб для моделювання методу пошуку оптимального шляху переміщення мобільної платформи серед невизначених перешкод має такі незалежні процеси:

- а) TraceRoute, що використовується для побудови траєкторії руху на основі даних про знайдені перешкоди;
- б) TimerProces – незалежний складний процес, який запускає наступні

функції:

- 1) сканування простору та пошук перешкод;
 - 2) візуалізація ефекту роботи сканеру оточуючого простору;
 - 3) перебудова віртуальної карти;
- в) переміщення цифрового двійника на віртуальній карті виробничого середовища за знайденим шляхом.

На рисунку 4.1 показано структуру програмного засобу.

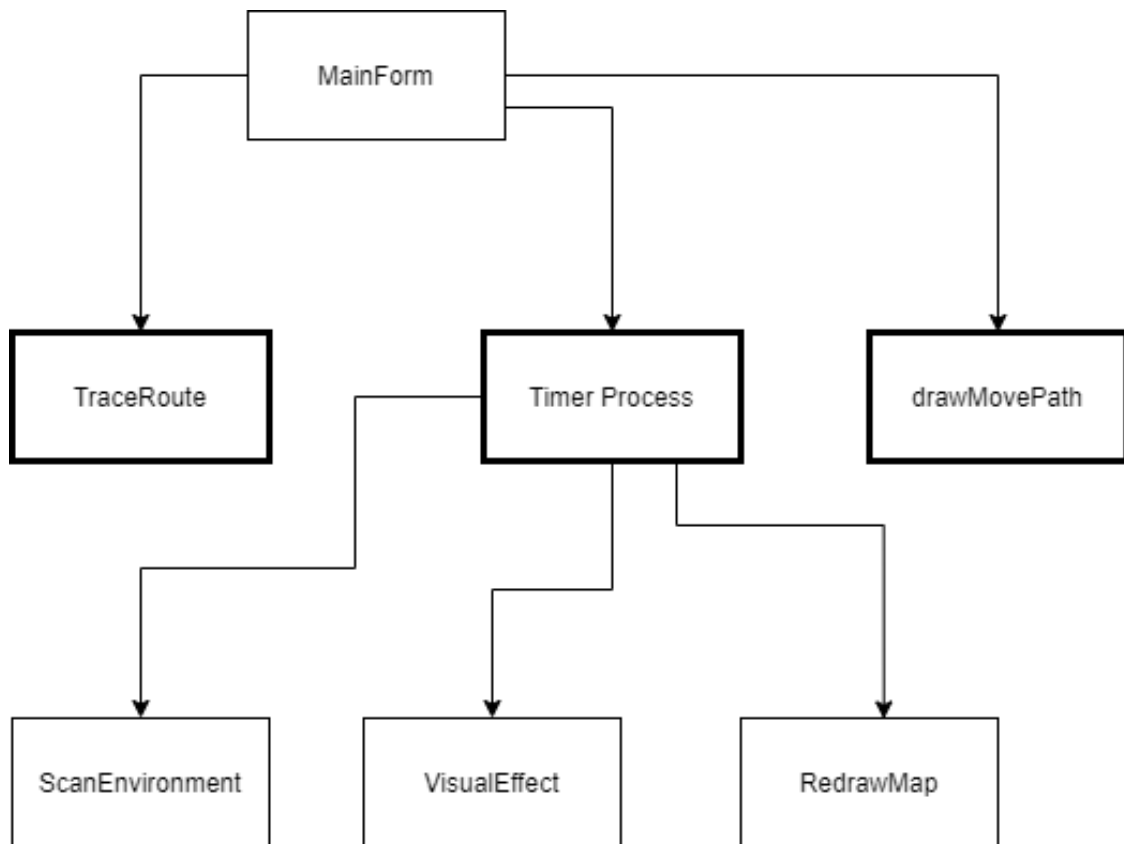


Рисунок 4.1 – Структурна схема програми для моделювання методу пошуку оптимального шляху переміщення мобільної платформи серед невизначених перешкод

Для візуалізації процесу сканування простору була розроблена функція `drawScan()`:

```
private void drawScan() {
    if (start.xIndex == 0 && start.yIndex == 0) return;
```

```

int xc = CellSize * start.xIndex - (int)(CellSize / 2) + start.xIndex;
int yc = CellSize * start.yIndex - (int)(CellSize / 2) + start.yIndex;
int scanLength = CellSize * 2 - 4;
Point pStart = new Point(xc, yc);
Point pScan_half = new Point((int)(pStart.X - (int)(scanLength / 2) *
Math.Cos(angleScan * Math.PI / 180)), (int)(pStart.Y - (int)(scanLength / 2) *
Math.Sin(angleScan * Math.PI / 180)));
setVisibleCell(pScan_half);
Point pScan = new Point((int)(pStart.X - scanLength * Math.Cos(angleScan
* Math.PI / 180)), (int)(pStart.Y - scanLength * Math.Sin(angleScan * Math.PI /
180)));
setVisibleCell(pScan);
Graphics g = Graphics.FromImage(pictureBox1.Image);
Pen mPen_line = new Pen(Color.DarkGray, 1);
g.DrawLine(mPen_line, pStart, pScan);
mPen_line.Dispose();
g.Dispose();
pictureBox1.Invalidate(); }

```

Дана функція реалістично моделює процес пошуку перешкоди, шляхом зображення на віртуальній карті пошукового проміню, який випромінюється з візуального сканеру, або лазерного лідару.

Коли промінь сканує поверхню та на його шляху з'являються предмети, через які не може проїхати мобільний робот, то їх координати заносяться до масиву `map[][]` за допомогою функції `setVisibleCell`:

```

private void setVisibleCell(Point pScan) {
int x = (int)(pScan.X / CellSize);
int y = (int)(pScan.Y / CellSize);
if ((x + 1 > 11) || (y + 1 > 11)) return;
map[x + 1, y + 1].isFind = true; }

```

Масив `map` зберігає всі точки простору. Кожна точка має властивості:

- вільна для переміщення;
- зайнята перешкодою;
- просканована, чи ні.

При виконанні процесу сканування робот послідовно відкриває точки карти та робить їх видимими, коли промінь проходить поз них.

Після знаходження нової перешкоди вона візуалізується (проявляється) на карті. Дану задачу вирішує функція `reDrawMap`:

```
private void reDrawMap() {
    drawClearMap();
    Point p = new Point();
    Color c = new Color();
    for (int i = 1; i <= MapWidth; i++)
        for (int j = 1; j <= MapHeight; j++) {
            if (!checkBox2.Checked)
                if (map[i, j].isFind == false) continue;
            p.X = (i - 1) * (CellSize + 1) + 2;
            p.Y = (j - 1) * (CellSize + 1) + 2;
            int index = (int)map[i, j].type + 1;
            c = groupBox3.Controls["label"+ index].BackColor;
            FillRegion(c, p); }
    drawStart(startCoord);
    drawFinish(finishCoord); }
```

Дана функція рисує на екрані ділянки простору виділені різним кольором:

- білий – ділянка не просканована;
- світло-сірий – ділянка просканована та вільна для проїзду;
- сірий – ділянка просканована та зайнята для проїзду;
- зелений – початкова точка;
- помаранчевий – кінцева точка маршруту.

Для поєднання всіх процесів розроблено функцію підрахунку модельного часу timer:

```
private void timer1_Tick(object sender, EventArgs e) {
    reDrawMap();
    if (start != nullCell && finish != nullCell) button9_Click(sender, e);
    angleScan += 5;
    if (angleScan > 360) {
        angleScan = 0;
        if (FindPath.Count > 1) {
            Point p = new Point(0, 0);
            p.X = (start.xIndex - 1) * (CellSize + 1) + 2;
            p.Y = (start.yIndex - 1) * (CellSize + 1) + 2;
            int index = (int)map[start.xIndex, start.yIndex].type + 1;
            Color c = groupBox3.Controls["label" + index].BackColor;
            FillRegion(c, p);
            if (numStep == 0) {
                Cell start_c = new Cell(start.xIndex, start.yIndex);
                MovePath.Add(start_c); }
            numStep++;
            map[FindPath[1].xIndex, FindPath[1].yIndex].numStep = numStep;
            Cell find_c = new Cell(FindPath[1].xIndex, FindPath[1].yIndex);
            MovePath.Add(find_c);
            Cell cel = new Cell(FindPath[1].xIndex, FindPath[1].yIndex);
            startCoord = new Point(FindPath[1].xIndex * CellSize - 20,
FindPath[1].yIndex * CellSize - 20);
            start = cel;
            if (start != nullCell)
                FillRegion(label4.BackColor, startCoord);
            drawStart(startCoord); } }
    drawScan();
}
```

```

if (start == finish) {
timer1.Enabled = false;
reDrawMap();
Application.DoEvents();
reDrawMap();
drawMovePath(); } }

```

Як можна бачити, ця функція виконує декілька задач. Перша задача – послідовна зміна куту зору скануючого проміню (`angleScan`). Кут змінюється з кроком 5 градусів. Цього досить для плавного зміщення зображення на екрані та для обробки всіх необхідних координат в площині ділянок на карті виробничого простору.

Наступна задача – візуалізація зміщення мобільного робота після завершення повного циклу сканування простору.

Третя задача – виклик процедури `pathFinder` та оновлення карти для візуалізації знайденого шляху.

Після завершення переміщення мобільної платформи та досягнення кінцевої точки таймер вимикається і викликається функція `drawMovePath` для візуалізації пройденого шляху.

Функція `drawMovePath` має наступний вигляд:

```

private void drawMovePath()
{
if (MovePath.Count == 0) return;
Point[] points = new Point[MovePath.Count];
int i = 0;
foreach (Cell c in MovePath) {
int xc = CellSize * c.xIndex - (int)(CellSize / 2) + c.xIndex;
int yc = CellSize * c.yIndex - (int)(CellSize / 2) + c.yIndex;
Point p = new Point(xc, yc);
points[i] = p; i++;
Graphics g = Graphics.FromImage(pictureBox1.Image);

```

```
g.SmoothingMode = System.Drawing.Drawing2D.SmoothingMode.  
AntiAlias;
```

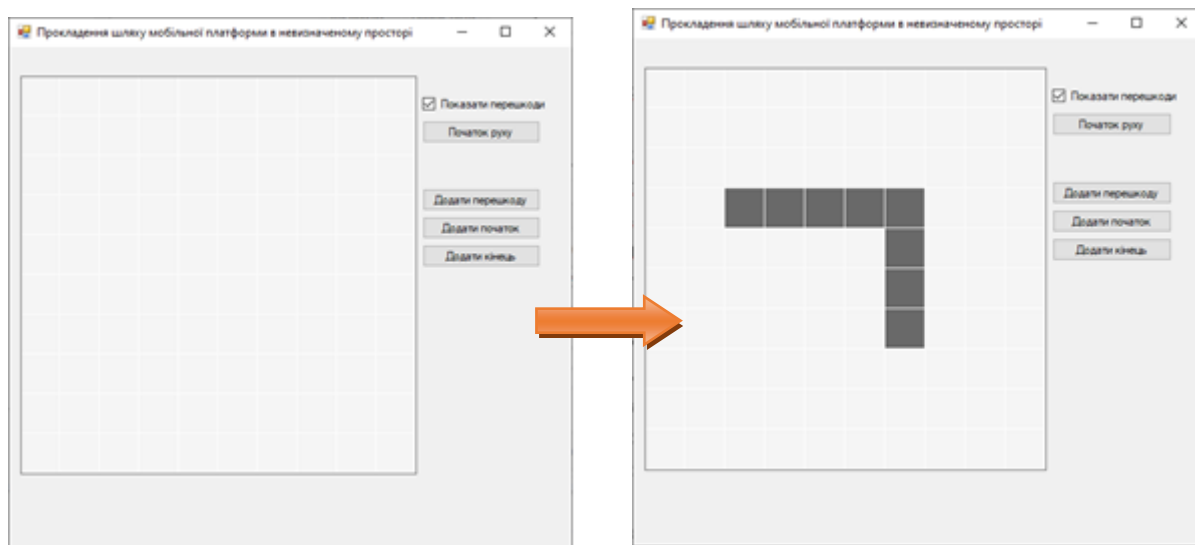
```
Pen mPen_line = new Pen(Color.DarkGray, 3);  
g.DrawCurve(mPen_line, points);  
mPen_line.Dispose();  
g.Dispose();  
pictureBox1.Invalidate(); }
```

Дана функція викликається після досягнення мобільною платформою кінцевої точки. Вона рисує на карті пройдений шлях роботом для візуалізації всіх його зміщень в процесі пошуку траєкторії.

Шлях рисується за даними, що зберігаються в масиві MovePath. В даний масив заносяться всі точки в яких знаходився мобільний робот процесі руху. Масив MovePath змінює функція зміщення мобільної платформи.

4.3 Тестовий запуск програми та проведення іспитів

При першому запуску програма має наступний вигляд, що показано на рисунку 4.2, а.



а – порожня карта

б – карта с перешкодами

Рисунок 4.2 – Підготовка до проведення тестування розробленого алгоритму

Спочатку поле для дослідження порожнє (рисунок 4.2, а). На цьому етапі треба завдати положення статичних перешкод, які не будуть змінюватись в процесі роботи обладнання.

Перешкода додається натисканням лівої кнопки миші на потрібній ділянці карти.

Таким чином створюється карта розміщення відомих перешкод, але які ще не відомі мобільному роботу (рисунок 4.2, б).

Інтерфейс програми має наступні органи керування (рисунок 4.3):

- перемикач візуалізації перешкод;
- кнопка старту експерименту;
- кнопка включення режиму додавання перешкоди;
- кнопка включення режиму додавання точки старту;
- кнопка включення режиму додавання кінцевої точки.

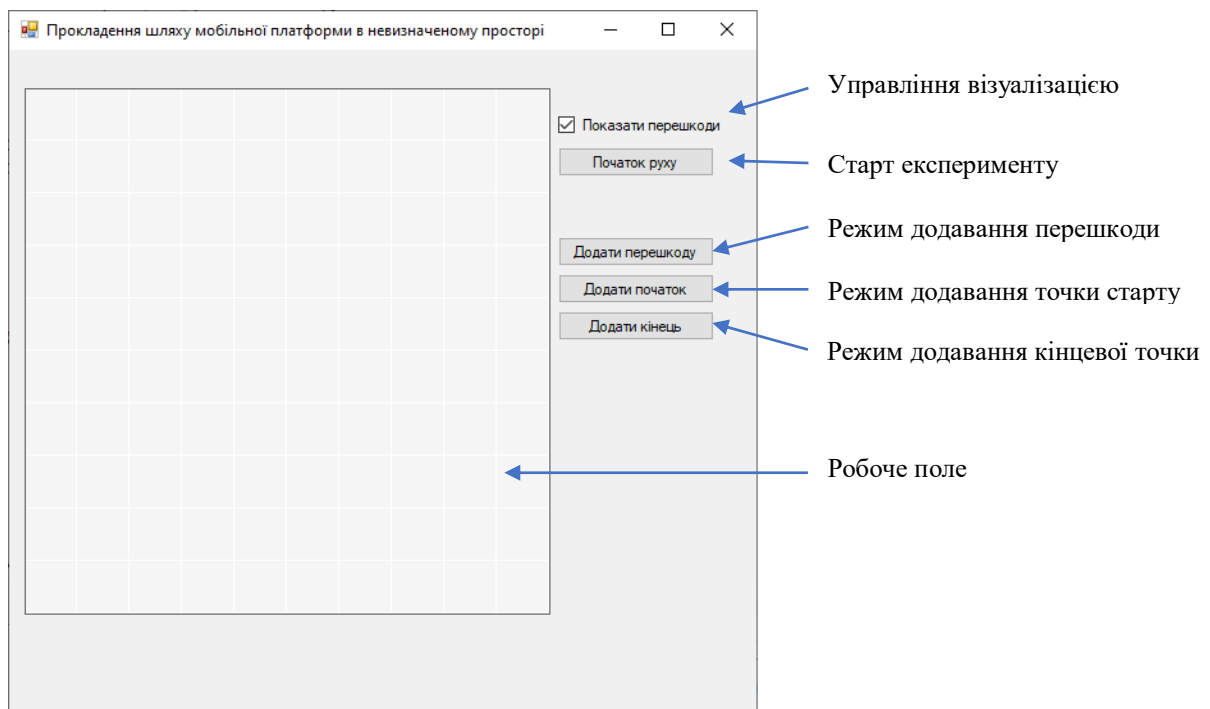


Рисунок 4.3 – Структура інтерфейсу програми

За допомогою перемикача візуалізації можна включити режим бачення мобільної платформи. В такому режимі можна спостерігати, як послідовно будується карта місцевості та прокладається шлях руху, враховуючи нові перешкоди. Режим можна перемикати в процесі роботи алгоритму в будь-який час. Після додавання на карту всіх необхідних ділянок, через які робот проїхати не може, оператор включає режим бачення мобільної платформи. Роботу програми в даному режимі показано на рисунку 4.4.

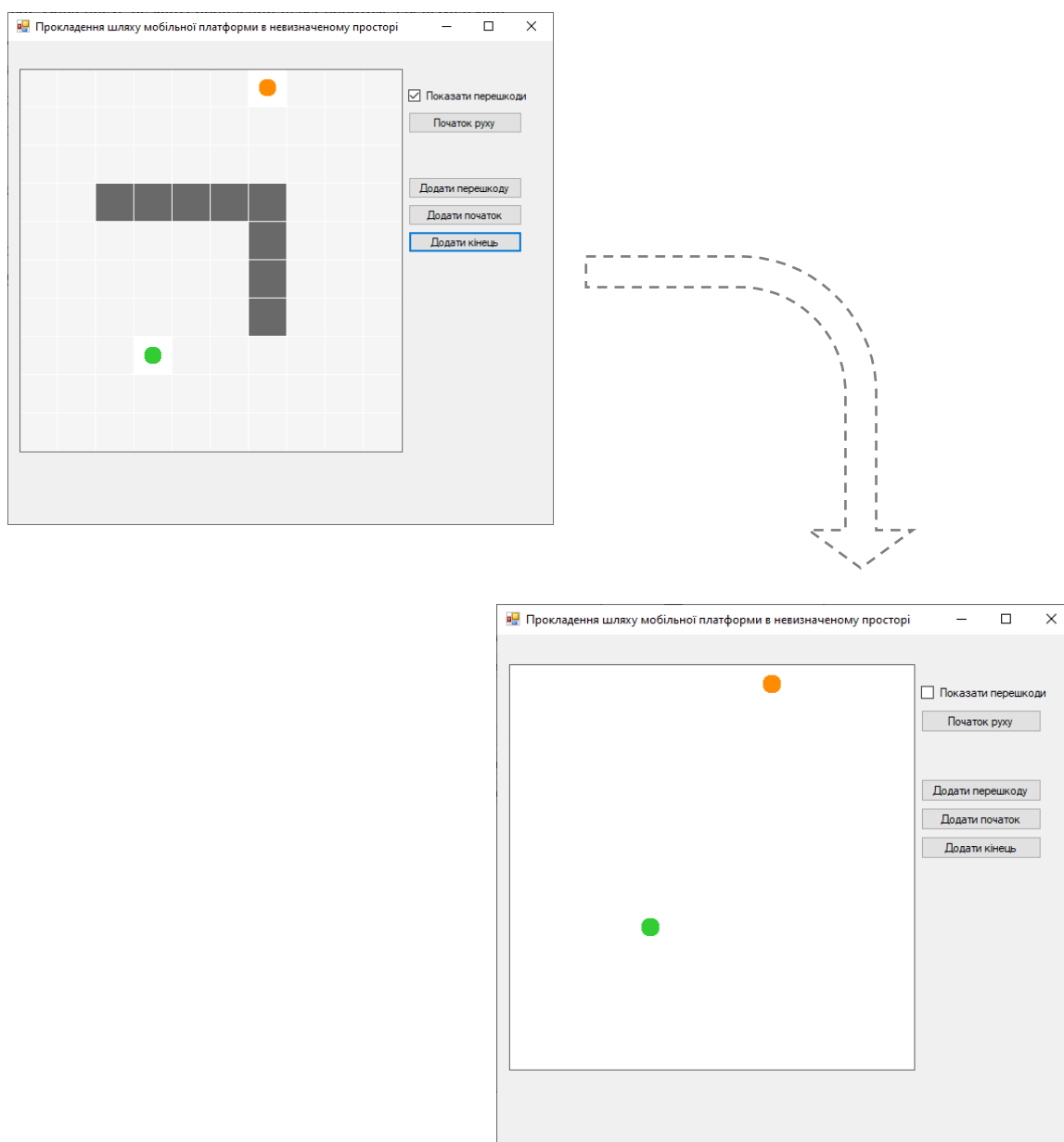


Рисунок 4.4 – Режим бачення мобільної платформи

Після натискання на кнопку «Початок руху», програма починає роботу в автоматичному режимі.

Завершиться робота автоматично, після досягнення кінцевої точки маршруту, або якщо подальший рух є неможливий.

Як можна бачити з рисунку 4.4 для мобільної платформи все поле є вільним для руху і обмежене лише рамками карти.

Після початку руху запускаються внутрішні незалежні процеси (побудова траєкторії руху, сканування простору та пошук перешкод, візуалізація ефекту роботи сканера оточуючого простору, перебудова віртуальної карти, переміщення цифрового двійника на віртуальній карті виробничого середовища за знайденим шляхом).

На рисунку 4.5 показано приклад роботи алгоритму на перших кроках роботи.

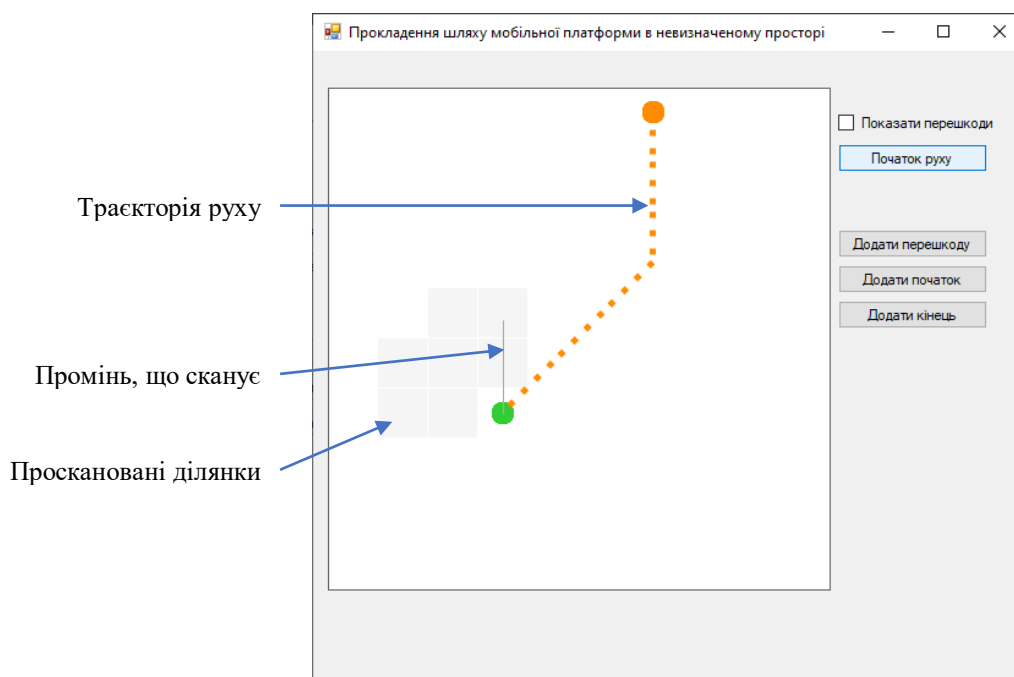


Рисунок 4.5 – Приклад роботи алгоритму на перших кроках роботи

З наведеного рисунку можна бачити, як працює процес сканування оточуючого простору. Промінь, що сканує, обертаючись відносно центру мобільної платформи, послідовно відкриває ділянки карти.

Поки ділянки не зайняті, вони мають світло-сірий відтінок. Коли буде знайдена непрохідна перешкода, вона буде показана темно-сірим кольором. Як було сказано раніше, в процесі роботи можна переключати режим візуалізації (робот - людина).

На рисунку 4.6 показано приклад переключення в інший режим (як бачить карту оператор).

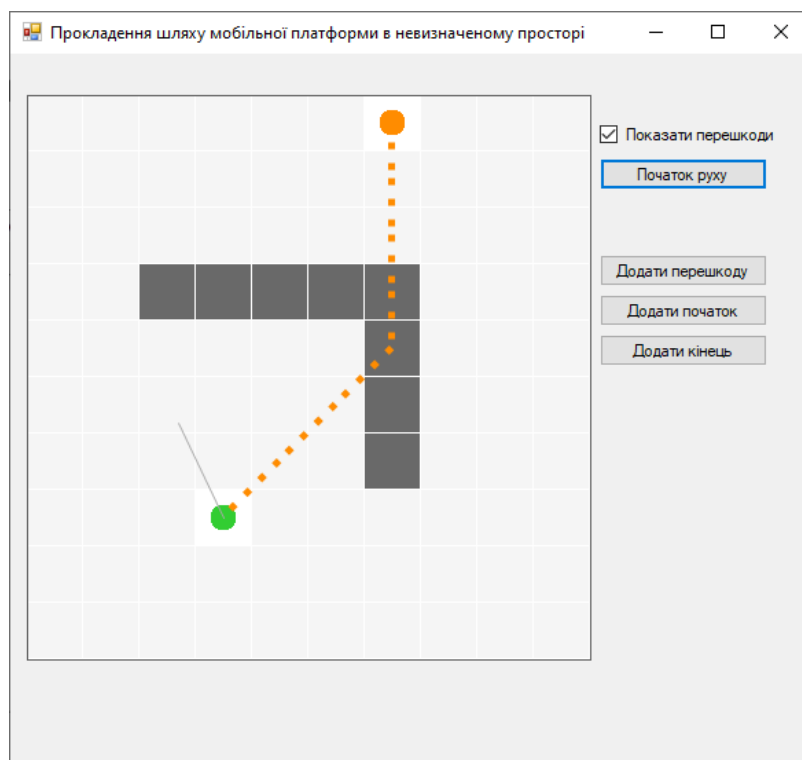


Рисунок 4.6 – Режим візуалізації «Показати перешкоди»

В даному режимі можна бачити всі перешкоди, навіть ті ще не були виявлені сканером мобільної платформи. Як можна бачити з рисунку 4.6 робот проклав шлях через перешкоди, тому, що ще не бачив їх за допомогою своїх сенсорів.

На рисунках 4.7-4.12 показана серія прикладів руху мобільної платформи та послідовного виявлення перешкод на кожному кроці при зміні стану оточуючого простору незалежний процес «TraceRoute» перебудовує траєкторію руху, а інший процес виконує зміщення мобільної платформи за новим курсом.

На рисунку 4.7 можна бачити, як платформа рухається вільним простором, слідкуючи за прокладеним шляхом. Точки траєкторії зберігаються в двовірному масиві `map [][]`. Зберігаються координати X та Y.

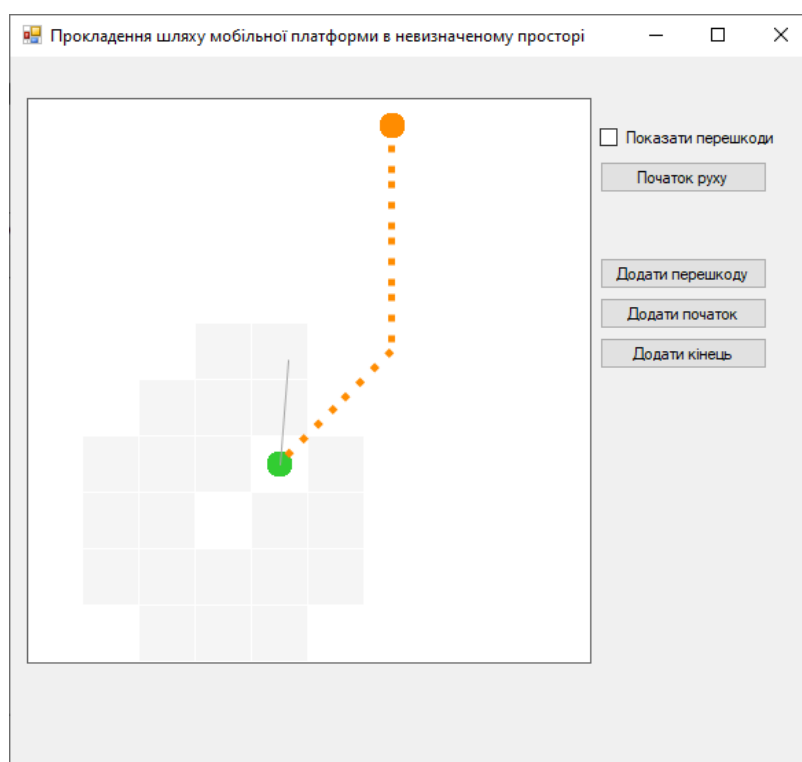


Рисунок 4.7 – Рух вільним простором

Перешкоди з'являються не всі відразу, а частками. Тому може бути така ситуація, що показана на рисунку 4.8, коли вся перешкода ще не видна і мобільна платформа все ще прокладає шлях через вільну для неї ділянку.

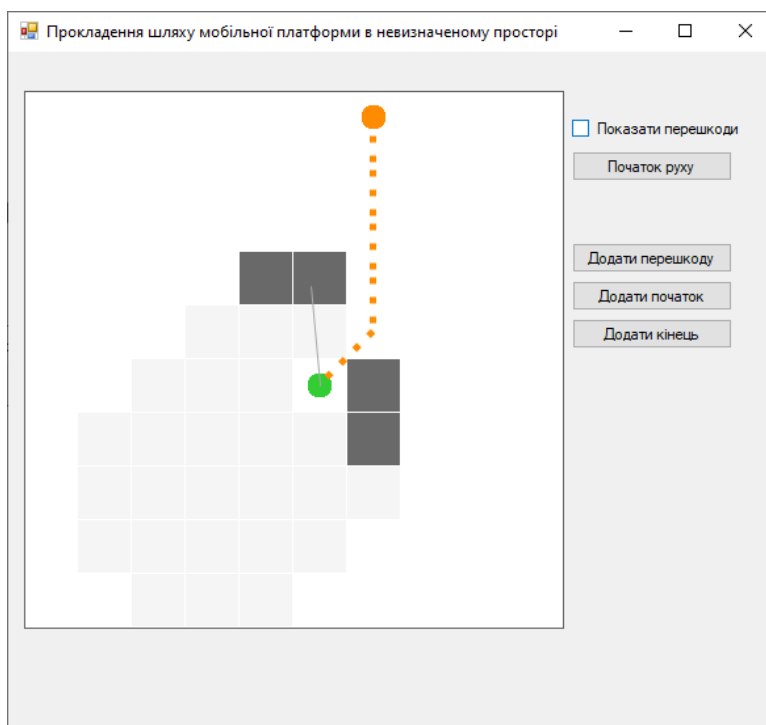


Рисунок 4.8 – Відкрита не вся перешкода

Коли біде виявлено, що вперед їхати вже не можна (перешкода перегороджує прохід) процес прокладання шляху в реальному часі перебудує траєкторію. Такий випадок показано на рисунку 4.9.

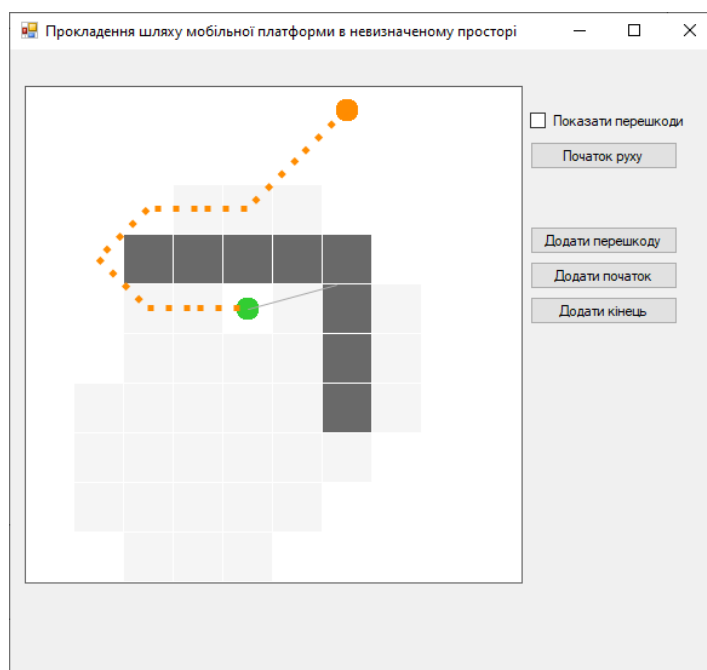


Рисунок 4.9 – Перебудова траєкторії руху

В процесі руху мобільна платформа може стикатися з появою рухомих перешкод. Наприклад, це може бути людина, або інший робот.

Такий випадок показано на рисунку 4.10.

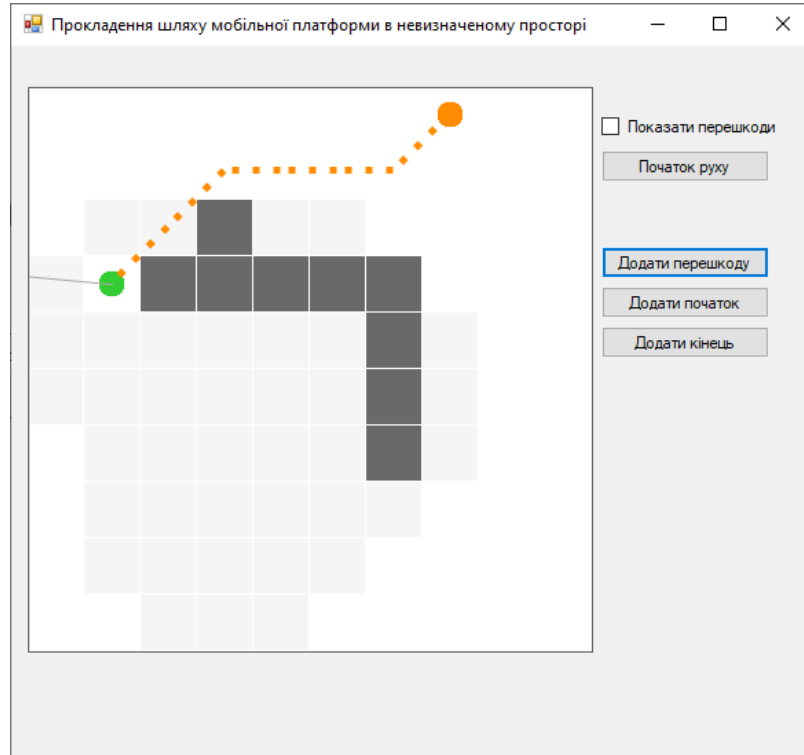


Рисунок 4.10 – Поява нової перешкоди на шляху руху платформи

В такому випадку знов буде перебудована траєкторія руху враховуючи межі оточуючого простору та розставлене і знайдене обладнання.

Якщо нові перешкоди заважатимуть продовженню руху в обраному напрямку, алгоритм може повністю перебудувати шлях переміщення (рисунок 4.11).

Вертаючись назад, мобільна платформа знову ж таки може знайти раніш не існуючи перешкоди.

На рисунку 4.12 показано приклад появи динамічних перешкод і нової перебудови траєкторії.

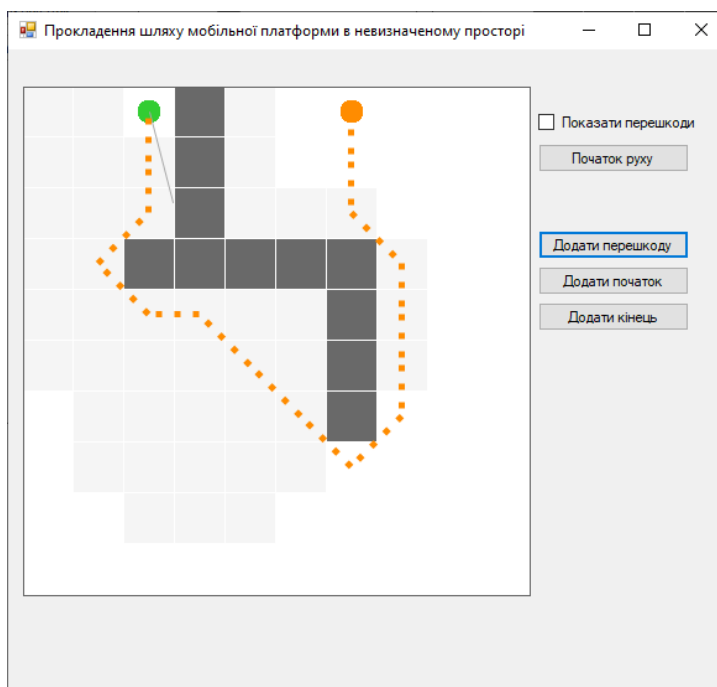


Рисунок 4.11 – Зміна напрямку руху

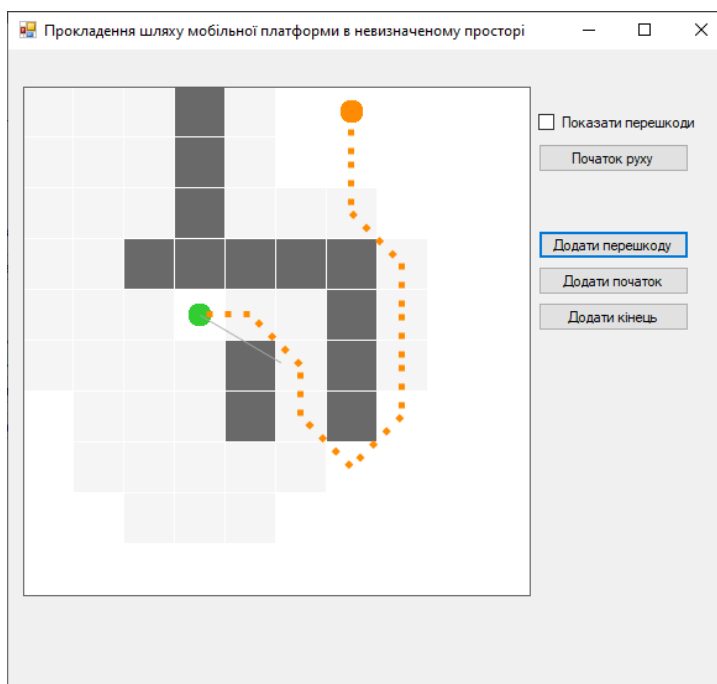


Рисунок 4.12 – Приклад появи динамічних перешкод і нової перебудови траєкторії

Після закінчення руху, коли робот досягне кінцевої точки, на екрані буде показано повний пройдений шлях (рисунок 4.13). На рисунку відображаються, як знайдені перешкоди так і всі точки пройденої траєкторії.

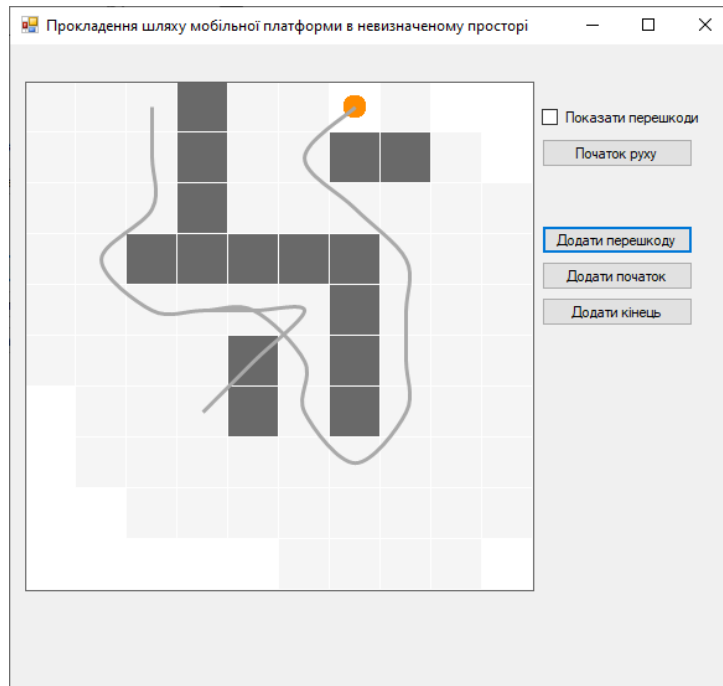


Рисунок 4.13 – Результат роботи алгоритму (Пройдена траєкторія)

Шлях може проходити через наявні перешкоди тому, що на момент попереднього проходу їх ще не було, а вони з'явилися при зміні траєкторії руху і враховані при прокладанні нового шляху.

На рисунку 4.14 показні статистичні дані експерименту.

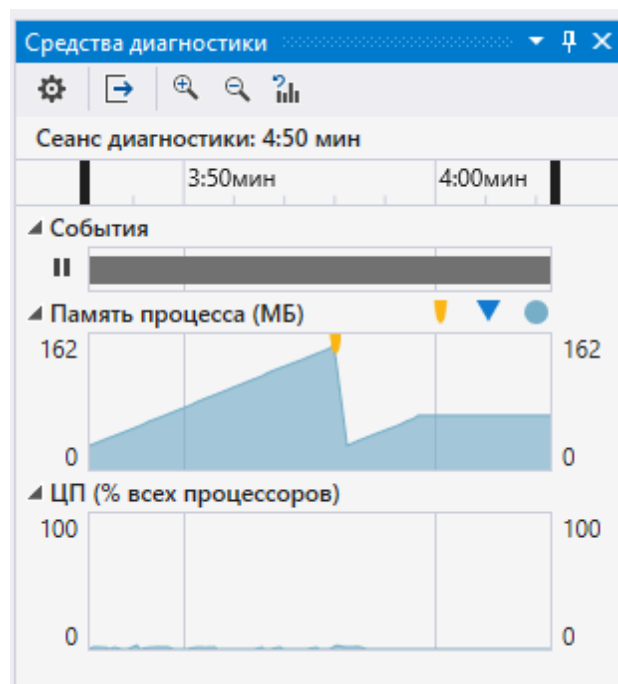
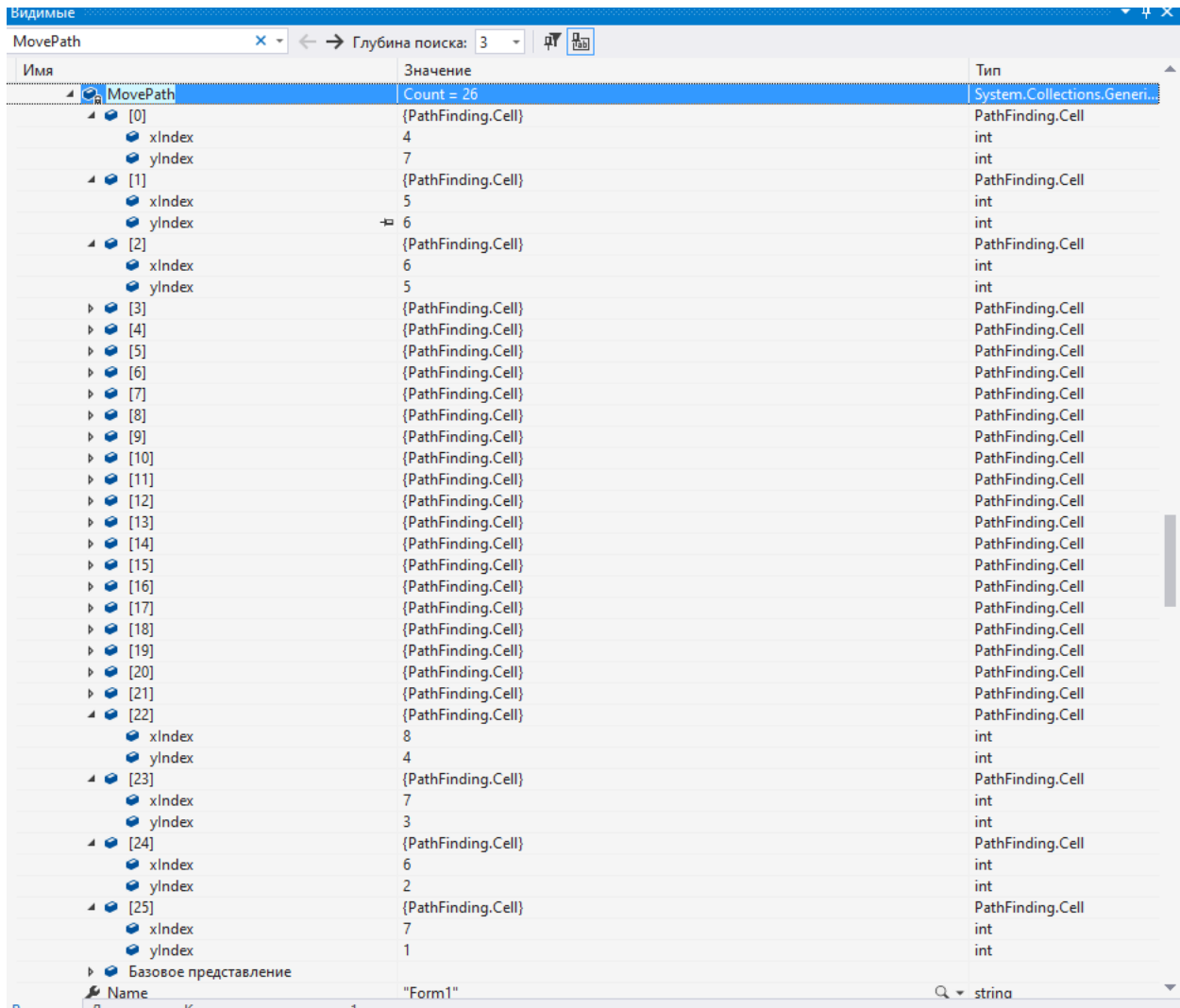


Рисунок 4.14 – Статистичні дані експерименту

З наведеного рисунку можна бачити, що остаточний час виконання задачі проїзду від початкової до кінцевої точок складає 4 хвилини (жовтий маркер над рівною позвехньою графіка).

На рисунку 4.15 показано результат роботи програми.



Имя	Значение	Тип
MovePath	Count = 26	System.Collections.Generi...
[0]	{PathFinding.Cell}	PathFinding.Cell
xIndex	4	int
yIndex	7	int
[1]	{PathFinding.Cell}	PathFinding.Cell
xIndex	5	int
yIndex	6	int
[2]	{PathFinding.Cell}	PathFinding.Cell
xIndex	6	int
yIndex	5	int
[3]	{PathFinding.Cell}	PathFinding.Cell
[4]	{PathFinding.Cell}	PathFinding.Cell
[5]	{PathFinding.Cell}	PathFinding.Cell
[6]	{PathFinding.Cell}	PathFinding.Cell
[7]	{PathFinding.Cell}	PathFinding.Cell
[8]	{PathFinding.Cell}	PathFinding.Cell
[9]	{PathFinding.Cell}	PathFinding.Cell
[10]	{PathFinding.Cell}	PathFinding.Cell
[11]	{PathFinding.Cell}	PathFinding.Cell
[12]	{PathFinding.Cell}	PathFinding.Cell
[13]	{PathFinding.Cell}	PathFinding.Cell
[14]	{PathFinding.Cell}	PathFinding.Cell
[15]	{PathFinding.Cell}	PathFinding.Cell
[16]	{PathFinding.Cell}	PathFinding.Cell
[17]	{PathFinding.Cell}	PathFinding.Cell
[18]	{PathFinding.Cell}	PathFinding.Cell
[19]	{PathFinding.Cell}	PathFinding.Cell
[20]	{PathFinding.Cell}	PathFinding.Cell
[21]	{PathFinding.Cell}	PathFinding.Cell
[22]	{PathFinding.Cell}	PathFinding.Cell
xIndex	8	int
yIndex	4	int
[23]	{PathFinding.Cell}	PathFinding.Cell
xIndex	7	int
yIndex	3	int
[24]	{PathFinding.Cell}	PathFinding.Cell
xIndex	6	int
yIndex	2	int
[25]	{PathFinding.Cell}	PathFinding.Cell
xIndex	7	int
yIndex	1	int
Базовое представление		
Name	"Form1"	

Рисунок 4.15 – Результат роботи програми

Отриманий кінцевий шлях переміщення робота складає 26 ділянок на карті.

Таким чином, розроблений метод пошуку оптимальної траєкторії руху протестовано на практиці і результат роботи імітаційної моделі показав його працездатність.

Оптимальність шляху гарантується властивістю алгоритму A^* та була доказана в роботах, що описують даний алгоритм. Але запропонований метод не завжди гарантує досягнення кінцевої точки. В випадку, коли динамічні перешкоди будуть повністю закривати всі можливі шляхи руху, алгоритм працювати не буде. Також, враховуючи ймовірний характер появи перешкод, час виконання завдання не може бути точно прогнозовано і він може змінюватись в процесі виконання завдання.

4.4 Розрахунок загальнообмінної вентиляції приміщення в якому проводиться розробка

Роботи в лабораторії відносяться до робіт категорії 1а – легка фізична робота, яка виконується сидячи.

Оптимальні норми мікроклімату згідно з ДСН 3.3.6.042-99: в холодний період року – температура 22-24 °C; відносна вологість 40-60 %; швидкість руху повітря не більше 0,1 м/с. У теплий період року – температура 23-25 °C; відносна вологість 40-60 %; швидкість руху повітря не більше 0,1 м/сек. Забезпечується за допомогою загальнообмінної вентиляції.

У приміщенні використовується суміщене освітлення: природне і штучне. Згідно ДБН В.2.5-28-2006 категорія зорових робіт, що проводяться в приміщенні – III В. Нормативні значення штучного освітлення $E = 200 \div 500$ лк, природного – КПО = 1,2 %. Штучне освітлення виконано як загальне, за допомогою світильників з люмінесцентними лампами.

Шум в приміщенні відповідає нормативним значенням – 50 дБА згідно з ДСН 3.3.6-037-99.

Місцева витяжна вентиляція передбачається для видалення шкідливих речовин та надлишкових тепловиділень в місці їх утворення. Вона попереджає їх поширення по приміщенню при найменшій витраті вентиляційного повітря, що підвищує ефективність вентиляції в цілому.

Установка місцевої витяжної вентиляції складається з приймачів забрудненого повітря – місцевих відсмоктувачів, повітропроводів, фільтра для очищення повітря (передбачається при значних масових викидах, у всіх випадках при видаленні пилу) і вентилятора.

Місцеві відсмоктувачі повинні бути максимально наближені до джерела шкідливих виділень і розташовані в напрямку їх природного руху.

Конструктивно місцеві відсмоктувачі можуть бути виконані у вигляді витяжних зонтів, панелей, шаф, бортових відсмоктувачів та ін.

Витяжні панелі встановлюють при проведенні процесів пайки, зварювальних робіт, вибивці лиття з піщаних форм та ін.

Вихідними даними для розрахунку витяжної панелі є:

- довжина $A = 0,4$ та ширина $B = 0,3$ джерела, m ;
- температура нагрітої поверхні джерела $t_u = 200\text{ }^{\circ}C$.

Панелі розміщують збоку від джерела шкідливих виділень на відстані від $b = 0$ до $b = 0,3\text{ м}$. Довжина панелі A повинна складати $0,48\text{ м}$.

Продуктивність витяжної панелі $Ln, m^3/год$, визначається за формулою:

$$Ln = c \cdot Q_k (h + B)^{5/3}, \quad (5.1)$$

де Q_k – кількість конвективного тепла, що виділяється джерелом, Bm ;

h – відстань від поверхні джерела до центру всмоктувальних отворів панелі, m (приймається $h = 0,4 \div 0,8$) $h = 0,6\text{ м}$;

$c = 228 [l / (h + B)]^{2/3}$ – коефіцієнт, що залежить від конструкції панелі і її розташування щодо джерела. Формула наведена для односторонньої панелі.

$$Q_k = \alpha_k \cdot F_i (t_i - t_{p.3}), \quad (5.2)$$

де α_k – коефіцієнт конвективної тепловіддачі, $Bm/m^2\text{ }^{\circ}C$;

$t_{p,3}$ – температура повітря в робочій зоні приміщення, $^{\circ}\text{C}$ (приймається відповідно до ДСН 3.3.6.042-99 для холодного періоду року);

F_i – площа робочої поверхні.

$$\alpha_K = 1,3 \cdot \sqrt[3]{t_H - t_{p,3}}, \quad (5.3)$$

$$l = 0,3 + 0,3 = 0,6 \text{ м}, \quad (5.4)$$

де l – сума ($\varphi + B$), м (відстань l приймається від $B \dots 2B$).

$$c = 228[0,6/(0,6+0,6)]^{2/3} = 228 \cdot 0,5^{2/3} = 143,63, \quad (5.5)$$

$$\alpha_K = 1,3 \cdot \sqrt[3]{200 - 23} = 7,3, \quad (5.6)$$

$$Q_k = 7,3 \cdot 0,16 \cdot (200 - 23) = 206,736 \text{ Вт}, \quad (5.7)$$

$$L_{\Pi} = 143,63 \cdot (206,736)^{1/3} \cdot (0,6 + 0,3)^{5/3} = 712,5 \text{ м}^3/\text{год}. \quad (5.8)$$

4.5 Висновки по четвертому розділу роботи

В четвертому розділу магістерської атестаційної роботи виконувались експериментальні дослідження практичного використання запропонованого методу прокладення оптимального шляху переміщення серед невизначених перешкод. Для цього використовувалось імітаційне (програмне) моделювання.

Була розроблена структурна схема програми для моделювання методу пошуку оптимального шляху переміщення мобільної платформи серед

невизначених перешкод. Програма написана мовою програмування C# в середовище Visual Studio.

Розроблений метод пошуку оптимальної траєкторії руху протестовано на практиці і результат роботи імітаційної моделі показав його працездатність. Оптимальність шляху гарантується властивістю алгоритму A* та була доказана в роботах, що описують даний алгоритм. Але запропонований метод не завжди гарантує досягнення кінцевої точки. В випадку, коли динамічні перешкоди будуть повністю закривати всі можливі шляхи руху, алгоритм працювати не буде. Також, враховуючи ймовірний характер появи перешкод, час виконання завдання не може бути точно прогнозовано і він може змінюватись в процесі виконання завдання.

ВИСНОВКИ

У роботі описані методи визначення траєкторії переміщення мобільного робота по площині серед перешкод. Згідно з цим роботи повинні уміти переміщатися по певних траєкторіях в автоматичному режимі. Для цієї мети використовується навігаційна система, яка визначає власні координати робота, планує траєкторію у нинішній момент часу і управляє його рухом.

Оскільки реальне середовище, в якому знаходиться робот, зазвичай містить рухливі перешкоди (люди, інші мобільні роботи), рух в ній по заздалегідь заданій траєкторії практично неможливий.

У роботі розглянуті рішення задачі управління рухом мобільного робота за наявності рухливих (динамічних) перешкод. У випадку з частково-лінійною траєкторією від початкової точки до цільової, МП може коригуватися в реальному часі з використанням алгоритму A^* . Система навігації мобільного робота відстежує його положення і обчислює координати поточної опорної точки, до якої повинен рухатися робот, уникаючи зіткнень з перешкодами (статичними і динамічними).

Суть запропонованого методу полягає в поєднанні методу пошуку оптимальної траєкторії руху, коли всі перешкоди визначені, та методу візуальної одометрії для визначення змін в навколишньому просторі. Поєднавши ці два методи ми отримуємо можливість постійного пошуку та аналізу розташування нових перешкод, що з'являються в оточуючому просторі, та швидкого реагування на знайдені зміни в реальному часі.

Програмний засіб для моделювання методу пошуку оптимального шляху переміщення мобільної платформи серед невизначених перешкод має такі незалежні процеси: побудова траєкторії руху на основі даних про знайдені перешкоди; сканування простору та пошук перешкод; візуалізація ефекту роботи сканеру оточуючого простору; перебудова віртуальної карти;

переміщення цифрового двійника на віртуальній карті виробничого середовища за знайденим шляхом.

Розроблений метод пошуку оптимальної траєкторії руху протестовано на практиці і результат роботи імітаційної моделі показав його працездатність. Оптимальність шляху гарантується властивістю алгоритму A^* та була доказана в роботах, що описують даний алгоритм. Але запропонований метод не завжди гарантує досягнення кінцевої точки. В випадку, коли динамічні перешкоди будуть повністю закривати всі можливі шляхи руху, алгоритм працювати не буде. Також, враховуючи ймовірний характер появи перешкод, час виконання завдання не може бути точно прогнозовано і він може змінюватись в процесі виконання завдання.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. ДСТУ 3008-2015. Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлення: Введ. 2015-22-06. – К.: Вид-во стандартів, 2016. – 26 с.

2. Методичні вказівки з «Розробки й оформлення магістерської атестаційної роботи» для студентів другого (магістерського) рівня вищої освіти галузі знань 15 Автоматизація та приладобудування за спеціальністю 151 Автоматизація та комп'ютерно-інтегровані технології освітні програми: «Автоматизоване управління технологічними процесами», «Комп'ютерно-інтегровані технологічні процеси і виробництва», «Комп'ютеризовані та робототехнічні системи» / Упоряд. І.Ш. Невлюдов, В.В. Косенко, В.В. Євсєєв. – Харків: ХНУРЕ, 2019. – 55 с.

3. Теслюк С.І. Аналіз методів локалізації рухомого об'єкта у виробничих приміщеннях / С.І. Теслюк, Д.В. Мамонько / Матеріали конференції «Автоматизація, електроніка та робототехніка. Стратегії розвитку та інноваційні технології». 2019. – С. 10-13.

4. R. Li, S. Wang, Z. Long, and D. Gu, “UnDeepVO: Monocular visual odometry through unsupervised deep learning,” in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), 2018, pp. 7286–7291.

5. Мамонько Д.В. Аналіз методів локалізації об'єкта в розподілених сенсорних мережах за допомогою акустичних джерел / Збірник студентських наукових статей «Automation and Developed of Electronic Devices» ADED-2018. – 2018. – № 2. – С. 29-32.

6. Мамонько Д.В. Аналіз методів керування роботизованою платформою на основі даних візуальної одометрії / Збірник студентських наукових статей «Automation and Developed of Electronic Devices» ADED-2019. – 2019. – № 1. – С. 114-117.

7. Демченко В.В. Геометричне моделювання і комп'ютерна графіка / В.В. Демченко, Є.В.Бородавка. – К.: КНУБА, 2010. – 288 с.

8. Н-мост и схема работы для управления двигателями [Електронний ресурс]. – Електрон. текстові дані. – Режим доступу: <https://arduino-master.ru/uroki-arduino/shema-raboty-n-mosta-dlya-upravleniya-dvigatelyami/> – 12.11.2020.

9. Мамонько Д.В. Розробка програмного для забезпечення керування траєкторією руху мобільної платформи / Збірник студентських наукових статей «Automation and Developed of Electronic Devices» ADED-2020. – 2020. – № 1. – С. 112-117.

10. RoboCraft (технология SLAM) [Електронний ресурс]. – Електрон. текстові дані. – Режим доступу: <http://robocraft.ru/blog/technology/724.html> – 12.11.2020.

11. Balasuriya B. Outdoor robot navigation using Gmapping based SLAM algorithm / B. Balasuriya, B. Chathuranga, B. Jayasundara, N. Napagoda, S. Kumarawadu, D. Chandima, A. Jayasekara / 2nd International Moratuwa Engineering Research Conference (Sri Lanka: Moratuwa). 2016 – pp 403.

12. GMapping [Електронний ресурс]. – Електрон. текстові дані. – Режим доступу: <http://wiki.ros.org/gmapping> – 15.10.2020.

13. Localize TurtleBot Using Monte Carlo Localization [Електронний ресурс]. – Електрон. текстові дані. – Режим доступу: <https://es.mathworks.com/help/nav/ug/localize-turtlebot-using-monte-carlo-localization.html>. – 17.11.2020.

14. Невлюдов І. Ш. Людино-машинний інтерфейс в технічних засобах автоматизації: Навчальний посібник / І. Ш. Невлюдов, О. І. Филипенко, Б. О. Шостак. – Харків : «ХТМТ», 2019. – 244 с.

15. Caruso D., Engel J., Cremers D. Large-scale direct slam for omnidirectional cameras //2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), September 28 - October 2. 2015, Hamburg. Germany. - Hamburg, 2015.-P. 141-148.

16. Nistrer D. Visual odometry for ground vehicle applications / D. Nistrer, O. Naroditsky, J. Bergen // Journal of Field Robotics. – 2006. – Vol. 23 (1). – P. 3-20.

17. Девятериков, Е.А. Использование данных визуального одометра для автономного возвращения мобильного робота в среде без фиксированных точек отсчета / Е.А. Девятериков, Б.Б. Михайлов // Экстремальная робототехника: Сборник докладов всероссийской научно-технической конференции. – СПб.: Изд-во «Политехника-сервис», 2015. – С. 356-361.

18. ORB-SLAM2 for Monocular, Stereo and RGB-D Cameras [Электронный ресурс]. – Электрон. текстові дані. – Режим доступу: <https://webdiis.unizar.es/~raulmur/orbslam/> – 17.11.2020.

19. Omnidirectional sensors for mobile robots / S. Blumenthal, D. Droschel, D. Holz, T. Linder, P. Molitor, H. Sunnann // Workshop Proceedings of SIMPAR 2008. Venice, Italy, November, 3-7, 2008. - Venice, 2008. - P. 414-425.

20. Caruso D. Large-scale direct slam for omnidirectional cameras // 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), September 28 - October 2, 2015, Hamburg, Germany. - Hamburg, 2015.-P. 141-148.

20. Rituerto A. Visual SLAM with an omnidirectional camera // 2010 20th International Conference on Pattern Recognition CPR2010, 23-26 August 2010, Istanbul, Turkey: Proceedings. - Istanbul, 2010. - P. 348-351.

21. An omni-directional vSLAM based on spherical camera model and 3D modeling / Guofeng Tong, Zizhang Wu, NingLong Weng, Wenbo Hou // Proceedings of the 10th World Congress on Intelligent Control and Automation. - Beijing, China, 2012.-P. 4551-4556.

22. Benefit of large field-of-view cameras for visual odometry / Z. Zhang, H. Rebecq, C. Forster, D. Scaramuzza // 2016 IEEE International Conference on Robotics and Automation (ICRA). - Stockholm, Sweden, 2016. - P. 801-808.

23. Kaess M., Dellaert F. Probabilistic structure matching for visual SLAM with a multi-camera rig // Computer Vision and Image Understanding. - 2010. - Vol. 114.-P. 286-296.

24. Carrera G., Angeli A., Davison A.J. SLAM-based automatic extrinsic calibration of a multi-camera rig // 2011 IEEE International Conference on Robotics and Automation. - Shanghai, China, 2011. - P. 2652-2659.

25. Davison A., Gonzalez Y., Kita N. Real-time 3D SLAM with wide-angle vision // IF AC Proceedings Volumes. - 2004. - Vol. 37 (8). - P. 868-873.

26. Основи наукових досліджень: Навч. посібник / І.Ш. Невлюдов, Ю.М. Олександров, А.О. Андрусевич, О.О. Чала. – Кривий Ріг: Криворізький коледж НАУ, 2019. – 396 с.

27. Scaramuzza D, Siegwart R. Appearance guided monocular omnidirectional visual odometry for outdoor ground vehicles // IEEE Transactions on Robotics. -2008. - Vol. 24 (5). - P. 1015-1026.

28. Панков, В. Д. Сравнение алгоритмов локализации ORB SLAM и LSD SLAM / В. Д. Панков, В. А. Шульга. – Текст : // Молодой ученый. – 2018. – № 27 (213). – С. 16-19. – URL: <https://moluch.ru/archive/213/51977/> (дата звернення: 13.11.2020).

29. Raul Mur-Artal, Juan D. Tardos, J. M. M. Montiel and Dorian Galvez-Lopez (DBoW2). ORB-SLAM2 – URL:https://github.com/raulmur/ORB_SLAM2 (дата звернення: 13.11.2020).