

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет Автоматики і комп'ютеризованих технологій  
(повна назва)

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

## КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти перший (бакалаврський)  
(рівень вищої освіти)

Розроблення автоматизованої системи визначення  
мінімальної траєкторії руху роботизованої платформи  
(тема)

Виконав:  
здобувач 4 року навчання, групи  
АКТАКІТ-21-1

Гліб ІЛЬЄНКОВ

Спеціальність 151 Автоматизація та  
комп'ютерно-інтегровані технології  
(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітня програма Автоматизація та  
комп'ютерно-інтегровані технології  
(повна назва освітньої програми)

Керівник доцент Наталія ДЕМСЬКА  
(посада, власне ім'я, прізвище)

Допускається до захисту  
Зав. кафедри КІТАР

(підпис)

Ігор Невлюдов  
(прізвище, ініціали)

2025 р.

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

«21» червня 2025 р.



Гліб ІЛЬЄНКОВ

# ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

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

ЗАТВЕРДЖУЮ:

Зав. кафедри КІТАР \_\_\_\_\_

(підпис)

« 28 » квітня 2025 р.

## ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Здобувачеві \_\_\_\_\_ Гльєнкову Глібу Олексійовичу \_\_\_\_\_  
(прізвище, ім'я, по батькові)

1. Тема роботи Розроблення автоматизованої системи визначення мінімальної траєкторії руху роботизованої платформи

затверджена наказом по університету від 19.05.2025 р. № 390 Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 15.06.2025 р.

3. Вихідні дані до роботи \_\_\_\_\_

3.1 Мова програмування Python, \_\_\_\_\_

3.2 Середовище програмування PyCharm \_\_\_\_\_

3.3 Карта 20x20 умовних одиниць \_\_\_\_\_

3.4 Алгоритм штучної мурашиної колонії \_\_\_\_\_

4. Перелік питань, що потрібно опрацювати в роботі \_\_\_\_\_

4.1 Аналіз технічного завдання, 4.2 Аналіз літератури за темою роботи, 4.3

Аналіз сучасних методів розроблення систем визначення планування шляху МР,

4.4 Розробка програмного забезпечення для автоматизації пошуку шляху роботизованій платформі

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Слайди у форматі Power Point у кількості 12 слайдів з розширенням .pptx

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

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

### КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз технічного завдання	8.04.2025	виконано
2	Аналіз літератури за темою роботи	10.04.2025	виконано
3	Аналіз сучасних методів розроблення систем визначення планування шляху МР	22.04.2025	виконано
4	Розробка програмного забезпечення для автоматизації пошуку шляху роботизованої платформи	22.04.2025	виконано
5	Оформлення пояснювальної записки	12.05.2025	виконано
6	Оформлення презентації	20.06.2025	виконано
7	Подання роботи на рецензію	22.06.2025	виконано
8	Подання роботи на підпис зав. кафедри	24.06.2025	виконано
9	Подання атестаційної роботи в ЕК	25.06.2025	виконано

Дата видачі завдання 28 квітня 2025 р.

Здобувач \_\_\_\_\_ Гліб ІЛЬЄНКОВ  
(підпис) (власне ім'я, прізвище)

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

## РЕФЕРАТ

Пояснювальна записка: 54 с., 21 рис., 24 джерела, 3 додатки.

### АВТОМАТИЗАЦІЯ, ПОШУК ШЛЯХУ, РОБОТИЗОВАНА ПЛАТФОРМА, АЛГОРИТМ

Об'єкт розробки – процес навігації мобільних роботизованих платформ у частково або повністю невідомому середовищі.

Предмет розробки – методи та засоби автоматизованого планування мінімальної траєкторії руху мобільної платформи з урахуванням статичних і динамічних перешкод.

Мета роботи – розробка та покращення програмного забезпечення для мобільної роботизованої платформи, яке забезпечує побудову карти середовища в реальному часі, локалізацію та побудову оптимального маршруту до цілі з обходом перешкод.

Методи дослідження – аналіз алгоритмів для визначення траєкторії руху роботизованої платформи.

В роботі створена програма для побудування маршруту руху роботизованої платформи, проведено випробування.

Розроблене програмне забезпечення може використовуватися на різних підприємствах де є необхідність у використанні роботизованої платформи та автоматизації пошуку шляху для цієї платформи.

## ABSTRACT

Explanatory note: 54 p., 21 fig., 24 sources, 3 appendices.

AUTOMATION, PATH FINDING, ROBOTIC PLATFORM,  
ALGORITHM

The object of development is the process of navigation of mobile robotic platforms in partially or completely unknown environments.

The subject of development is methods and tools for automated planning of the minimum trajectory of a mobile platform considering static and dynamic obstacles.

The purpose of the work is to develop and improve software for a mobile robotic platform that enables real-time environment mapping, localization, and optimal path planning with obstacle avoidance.

Research methods are the analysis of algorithms for determining the trajectory of movement of a robotic platform.

In the work, a program for building a route for the movement of a robotic platform was created, and tests were conducted.

The developed software can be used at various enterprises where there is a need to use a robotic platform and automate the search for a path for this platform.

## ЗМІСТ

Вступ.....	6
1 Аналіз сучасних методів розроблення систем визначення планування шляху МР .....	9
1.1 Методи планування шляху робота .....	14
1.2 Класичні підходи.....	16
1.2.1 Дорожня карта (графік видимості; діаграма Вороного) .....	16
1.2.2 Розкладання комірки (CD) .....	18
1.2.3 Штучні потенційні поля (APF) .....	19
1.3 Аналіз технічного завдання .....	21
2 Алгоритмічне та апаратне забезпечення автоматизованої системи .....	23
2.1 Компоненти системи.....	23
2.2 Опис архітектури системи.....	24
2.3 Опис модулів .....	27
2.4 3D картографування.....	31
3 Розробка програмного забезпечення для автоматизації пошуку шляху роботизованої платформи .....	36
3.1 SLAM на основі мультисенсорного злиття .....	36
3.2 Опис алгоритму .....	37
3.3 Планування шляху для мобільних роботів.....	38
3.3.1 Глобальне планування шляху на основі вдосконаленого алгоритму мурашиної колонії.....	38
3.3.2 Планування місцевих доріжок .....	41
3.3.3 Планування шляху в поєднанні з покращеним алгоритмом колонії мурах та алгоритмом динамічного вікна.....	42
3.4 Практична реалізація .....	43
3.4.1 Основні функції які були реалізовані .....	44
3.5 Забезпечення безпечних умов праці при розробці автоматизованої	

системи .....	48
Висновки. ....	50
Перелік джерел посилання .....	51
Додаток А Апробація результатів роботи .....	55
Додаток Б Код програми .....	57
Додаток В Демонстраційний матеріал.....	67

## ВСТУП

В даний момент на різних підприємствах зростає кількість роботизованих платформ відповідно до цього зростає необхідність у різних системах руху для цих платформ. Автоматизація цього процесу дозволяє знизити вплив людського фактора: коли роботизована платформа має можливість самотійно орієнтуватися у просторі і оператору не потрібно його постійно направляти, це дозволяє зменшити навантаження на людину а також це дозволяє економити ресурси та час. Роботизована платформа повинна вміти самотійно знаходити шлях, обходити перешкоди та прокладувати найбільш оптимальний маршрут до цілі [1].

Основною проблемою при плануванні руху є невизначеність, властива управлінню та розпізнаванню робота. Включення невизначеності в планування може підвищити якість обчислених планів, що призводить до набагато надійнішої роботи робота. Планування в умовах невизначеності дозволило досягти значного прогресу за останні кілька років для мобільних роботів, але все ще значною мірою ігнорується для мобільних маніпуляторів. Одна з ключових причин може полягати в тому, що більшість робіт зазвичай використовують консервативний підхід, який полягає в тому, щоб скласти руку до певної «безпечної домашньої конфігурації» і розглядати мобільний маніпулятор по суті як мобільного робота, де розроблені для них планувальники можуть бути безпосередньо використані для боротьби з невизначеністю.

Однак це дуже обмежувальний підхід, який не працює в досить поширених сценаріях, коли мобільний маніпулятор не може рухатися від початку до мети без зміни конфігурації руки. Планувальники, які враховують невизначеність для мобільних роботів, не мають прямого відношення до мобільних маніпуляторів у таких сценаріях, оскільки врахування лише базової пози, невизначеності не є достатнім, оскільки це не гарантує безпечнішого

шляху. Тому, щоб отримати більш безпечний шлях для всього мобільного маніпулятора, слід застосовувати більш комплексний підхід, наприклад, крім невизначеності базової пози, необхідно враховувати також вплив цієї невизначеності на рухи маніпулятора [1].

Розробку надійного планувальника мобільних маніпуляторів важко реалізувати для додатків у реальному часі, якщо не вирішити проблему обширних обчислень, пов'язаних із плануванням мобільних маніпуляторів. Аспект реального часу особливо важливий, особливо в невідомих середовищах, що передбачає повторюване та перемежоване перепланування та дослідження (за допомогою датчиків) навколишнього середовища.

Об'єкт розробки – процес навігації мобільних роботизованих платформ у частково або повністю невідомому середовищі.

Предмет розробки – методи та засоби автоматизованого планування мінімальної траєкторії руху мобільної платформи з урахуванням статичних і динамічних перешкод.

Мета роботи – розробка та покращення програмного забезпечення для мобільної роботизованої платформи, яке забезпечує побудову карти середовища в реальному часі, локалізацію та побудову оптимального маршруту до цілі з обходом перешкод.

Методи дослідження – аналіз алгоритмів для визначення траєкторії руху роботизованої платформи.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- провести аналіз матеріальної бази, технічного та методичного забезпечення бази практики, використаних під час проходження практики та підготовки кваліфікаційної роботи;

- реалізувати програмну частину автоматизованої системи визначення мінімальної траєкторії руху автоматизованої платформи;

- оформити звіт згідно [2-4];

- проведені дослідження відповідають цілям сталого розвитку (ЦСР) 9.

## 1 АНАЛІЗ СУЧАСНИХ МЕТОДІВ РОЗРОБЛЕННЯ СИСТЕМ ВИЗНАЧЕННЯ ПЛАНУВАННЯ ШЛЯХУ МР

Робот – це машина, здатна реагувати на навколишнє середовище та збирати інформацію (з датчиків) про навколишнє середовище, щоб безпечно виконувати різні типи завдань. Автономні роботи можуть працювати та пересуватися самостійно, без безпосереднього контролю людини. На відміну від роботів, які використовуються на виробничих підприємствах, де середовище жорстко контролюється та повністю відоме, автономних роботів не завжди можна запрограмувати на виконання попередньо визначених дій, оскільки неможливо передбачити різні ситуації, з якими робот може зіткнутися.

Крім того, середовище може бути непередбачуваним або змінюватися з часом, що означатиме, що рухи робота повинні бути швидко змінені онлайн [5]. До застосування автономних роботів належать кур'єри в лікарнях, охоронці, військове спостереження, аерокосмічні дослідження, моніторинг навколишнього середовища, прибиральники та газонокосарки. Загалом, одним із найважливіших застосувань є використання автономних мобільних роботів у небезпечних зонах, таких як мінні поля або всередині атомних станцій. Завдяки властивій мобільним роботам автономності необхідно подолати низку проблем, включаючи невизначеність, помилки зондування, планування, навчання, надійність і відповіді в реальному часі [6]. Роботи повинні динамічно реагувати на зміни та адаптуватися до навколишнього середовища.

Зародження дисципліни роботизованого планування шляху можна віднести до середини 1960-х років. Проблема планування шляхів для роботів, широко відома як планування шляху робота (РПП) (RPP – Robot path planning), є критичною проблемою в області навігації мобільних роботів. Мета полягає в тому, щоб визначити мінімальний шлях, вільний від будь-яких зіткнень у визначеному середовищі, починаючи з заздалегідь визначеного місця й

закінчуючи в цільовому пункті призначення. Загалом існує багато способів, за допомогою яких робот переміщується до місця призначення, але найкращий маршрут насправді вибирається відповідно до набору правил. Найпоширенішими критеріями є найкоротша відстань, найменша кількість використаної енергії або найкоротший час із найкоротшою відстанню.

Оскільки метою планування шляху є пошук найкоротшого шляху з урахуванням обмежень, таких як задане середовище та рух без зіткнень, це можна розглядати як задачу оптимізації [7].

Карти, розташування та навігація для мобільних роботів можуть використовуватися за допомогою моделей середовища. Однак об'єм даних у цих моделях має бути керованим і відповідати вимогам обчислень у реальному часі. Дуже складною проблемою в області робототехніки є планування шляхів роботи.

Головна мета – знайти маршрут від початку до кінця, не зачепивши нічого. При вирішенні задачі навігації робота необхідно враховувати три ключові питання: ефективність, безпека та точність. Ефективність вимагає, щоб алгоритм не змушував робота робити додаткові кроки або багаторазово зупинятися й повертатися, що витрачає час і енергію, щоб визначити найкоротшу відстань за найкоротший проміжок часу.

Безпека та точність залежать від знаходження шляху, який ні про що не зачепить, залишаючись поруч із цим шляхом [7]. Проблеми навігації робота можна розділити на локалізацію, планування шляху, керування рухом і когнітивне відображення, як показано на рис. 1.1. Планування шляху є, мабуть, найважливішим аспектом навігації робота [8].

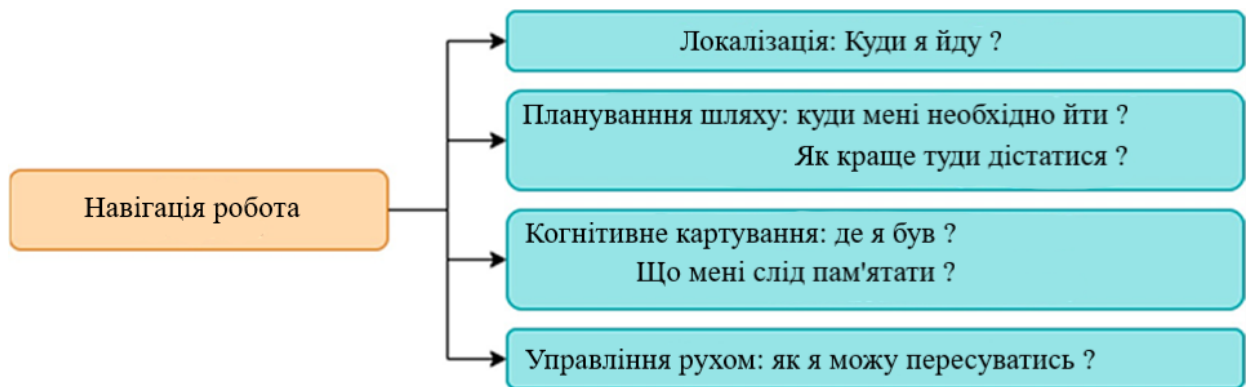


Рисунок 1.1 – Проблема навігації робота

Залежно від того, де вони розташовані [8], РПП поділяються на одну з двох категорій:

- РПП з фіксованими перешкодами в статичному середовищі.
- РПП, коли стикаються як з нерухомими, так і з рухомими перешкодами в динамічному середовищі.

Кожну з цих двох категорій можна додатково розділити на підгрупу. Глобальне планування шляху, коли роботи в середовищах можуть планувати свої загальні шляхи (офлайн) до того, як вони почнуть рухатися, оскільки вони мають повну інформацію про нерухомі перешкоди, а траєкторія рухомих перешкод відома заздалегідь.

Оскільки налаштувати нову карту складно, вартість зміни навколишнього середовища в глобальній навігації дуже висока, особливо в динамічному середовищі. Неможливість спланувати місцевий маршрут із повним знанням про навколишнє середовище є одним із обмежень планування місцевого маршруту.

Рухаючись у навколишньому середовищі, мобільний робот використовує датчики для збору інформації (онлайн). Робот повинен кілька разів переходити на нове місце, відчувати навколишнє середовище, оновлювати карту та планувати подальші дії, щоб створити карту свого оточення.

Планування локального шляху стикається з рядом проблем, включаючи

помилки зчитування датчиків, обмеження датчиків, змінні умови навколишнього середовища, динаміку робота, оцінку місця розташування та обмеження руху робота [6]. У надзвичайно складних і великих середовищах локальне планування шляху може бути не в змозі знайти шлях до цілі. Здебільшого це відбувається через нездатність датчиків надати достатньо даних, щоб відправити робота в потрібне місце [7].

Базова позиція робота, його обертання, а також обертання або трансляція всіх його ланок називаються конфігурацією робота. Набір усіх можливих конфігурацій становить конфігураційний простір або с-простір, і він вимагає кількох елементів:

- 1) опис форми робота, наприклад, чи є у нього ноги, колеса чи взагалі немає кінцівок (пересування);
- 2) пояснення середовища роботи, включаючи його геометрію (2- або 3-вимірний робочий простір);
- 3) середовище повинно мати початкову та цільову конфігурацію, між якими має бути спланований шлях робота.

Для опису конфігурації робочого простору робота використовується ряд параметрів. Наприклад, два параметри, які зазвичай називають  $x$  і  $y$ , можна використовувати для опису конфігурації робота, переводячи його в двовимірний робочий простір. Перешкоди в робочому просторі робота не дозволяють використовувати деякі конфігурації [8].

Картографування – це процес побудови моделі середовища. Щоб зробити карту, яка була б достатньо точною, щоб показати, що робот побачить на своєму шляху, рельєф має бути показаний точно. Для представлення навколишнього середовища існують різні типи карт. Деякі з них використовують сітку з клітинками, які або порожні, де робот може вільно переміщатися, або зайняті, щоб представляти перешкоду, як показано на рис. 1.2. Інші використовують двовимірні (або тривимірні) прямокутні чи квадратні робочі області, де перешкоди є багатокутниками, як показано на рис. 1.3.

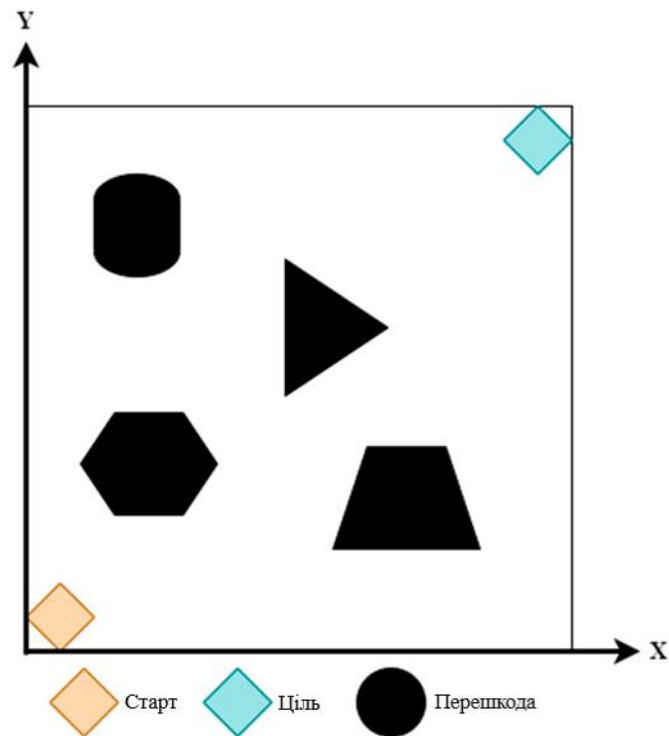


Рисунок 1.2 – Двовимірна карта навколишнього середовища

Багато дослідників представили оглядову статтю про навігацію мобільних роботів [9]; однак цих опитувань недостатньо для поглибленого аналізу окремих навігаційних методів. Цей запропонований опитувальний документ про навігацію мобільних роботів має на меті виявити прогалини в дослідженнях і обсяг інновацій у певній галузі. Він забезпечує поглиблений аналіз індивідуального алгоритму для статичного середовища, динамічного середовища за наявності рухомої перешкоди та цілі, аналіз моделювання, експериментальний аналіз, навігацію кількома мобільними роботами, гібридизацію з іншими інтелектуальними методами та його застосування до тривимірного (3D) середовища.

Опитування також підкреслює відмінності між класичними та реактивними підходами на основі їх ефективності та застосування в конкретних середовищах, таких як повітря, земля, підводне, промислове та небезпечне середовище. У [9] підхід обмежується лише тривимірним плануванням шляху, тоді як [10] розглядає підходи до планування шляху для кількох роботів і стратегії прийняття рішень для різних типів роботів,

включаючи повітряних, наземних і підводних роботів [11] фокусується на багатоцільових алгоритмах оптимізації для планування шляху мобільного робота.

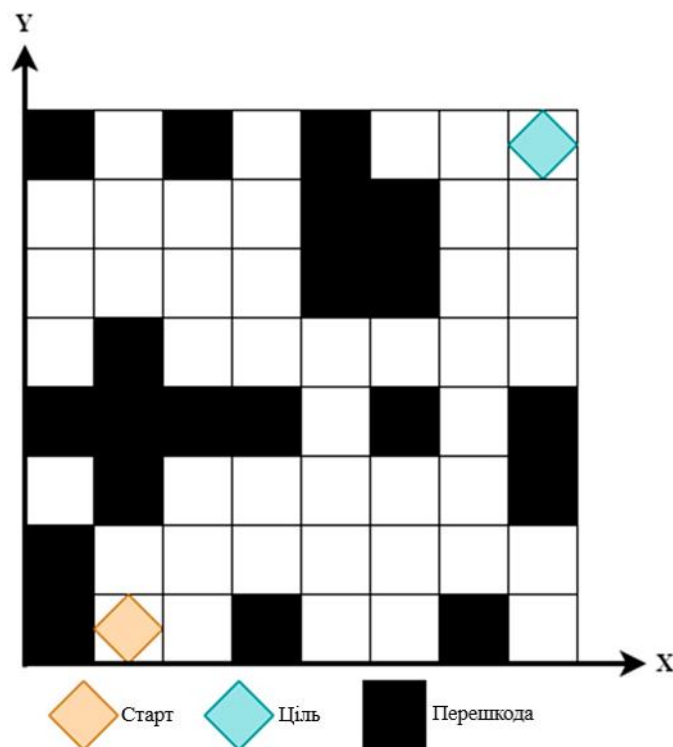


Рисунок 1.3 – Сітка карти

### 1.1 Методи планування шляху робота

Шлях робота повинен бути оптимізований відповідно до певних критеріїв, що робить питання планування шляху типом проблеми оптимізації. Вивчення алгоритмів оптимізації привернуло велику увагу дослідників протягом останніх кількох десятиліть. Два типи методів і алгоритмів оптимізації – детерміновані та стохастичні [12]. Стохастичні методи є більш вправними у виявленні глобальних оптимальних рішень для різних цільових функцій, тоді як детерміновані методи покладаються на математичні властивості проблеми.

І навпаки, стохастичні методи не залежать від математичних атрибутів конкретної функції, як зазначено в посиланні [13]. Тим не менш, початковий підхід демонструє певні обмеження, включаючи залежність від градієнтів,

сприйнятливість до локальних оптимумів, неефективність пошуку у великомасштабних просторах і нездатність звертатися до дискретних функцій.

Вважається, що методи, засновані на стохастичних процесах, мають більший ступінь зручності для користувача. Стохастичні методи необхідні через складність багатьох проблем реальної оптимізації, особливо коли йдеться про оптимізацію недиференційованих, мультимодальних і дискретних складних функцій – було виявлено, що ці алгоритми перевершують класичні або градієнтні підходи. Ці натхненні природою парадигми в даний час широко застосовуються в багатьох галузях техніки [7]. Ці методи виявилися надійними та потужними методами пошуку.

Протягом багатьох років численні вчені та дослідники пропонували різноманітні методології навігаційних підходів. Шляхи, якими мобільний робот знаходить свій шлях, можна умовно розділити на дві групи: класичні та евристичні (рис. 1.4).

Класичні методи мають низку недоліків, які роблять їх неефективними в реальних програмах, включаючи проблему високої складності часу у високих розмірах і явище захоплення локальними мінімумами, які є серйозними проблемами в цій галузі. Евристичні підходи перевершили звичайні підходи та набули широкої популярності в результаті NP-твердості проблеми РР. Евристичні методи також стають все більш популярними завдяки їх успіху у вирішенні проблем, пов'язаних із багатовимірністю, складними робочими просторами та локальними мінімумами [10].

У наступному підрозділі дослідження ми розглядаємо групу наукових внесків, які використовували класичні та евристичні підходи для пошуку найкращого планування шляху з однією або кількома цілями.

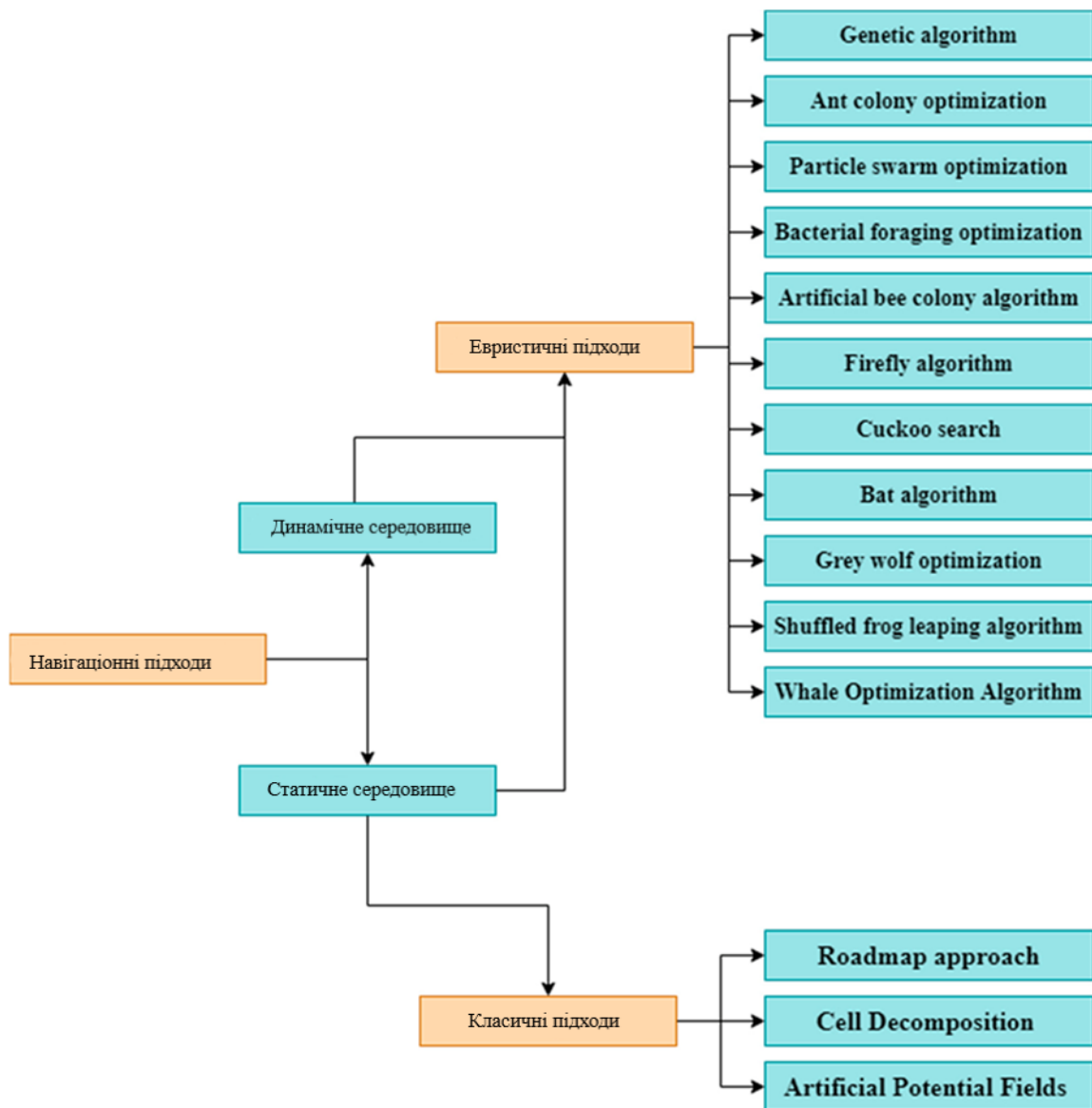


Рисунок 1.4 – Навігаційні підходи

## 1.2 Класичні підходи

До пошуку найкращих одно- або багатоцільових шляхів використовуються три класичні підходи: дорожня карта; розкладання комірки; штучні потенційні поля.

### 1.2.1 Дорожня карта (графік видимості; діаграма Вороного)

Підхід дорожньої карти (RM) передбачає переформулювання, скорочення або відображення вільного С-простору на одновимірний робочий

простір, як показано на рис. 1.5.

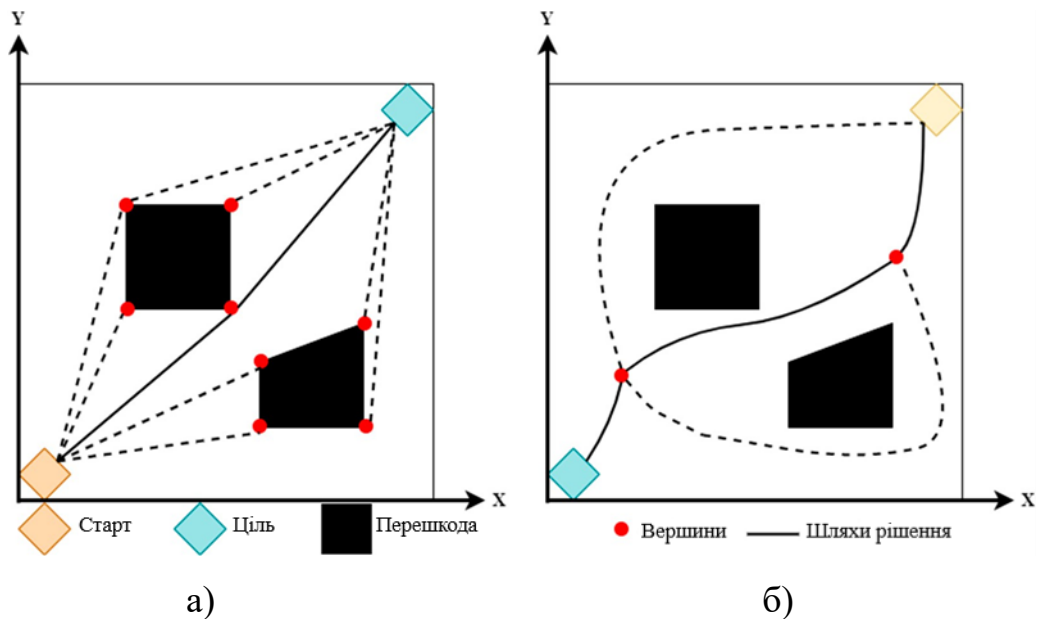


Рисунок 1.5 – Графік видимості (а); Діаграма Вороного (б)

В [10] досліджує застосування RM для навігації автономних мобільних роботів. На етапі планування простір конфігурації робота генерується випадковим чином, що призводить до певної кількості вузлів. Потім створюється дорожня карта шляхом з'єднання пар вузлів прямою лінією, при цьому слідкуючи за тим, щоб шлях не перетинався з жодними перешкодами. На основі результатів, отриманих на етапі планування, на етапі запити розробляється шлях, що з'єднує початкову та цільову конфігурації. Було впроваджено вдосконалення для ефективного планування шляху діаграми Вороного з метою оптимізації її продуктивності та пом'якшення проблем, таких як різкі повороти та розширені цикли, як цитується в посиланні [12]. Оптимальний маршрут визначається за допомогою використання гібридного підходу, який об'єднує графік видимості, діаграму Вороного та техніку потенційного поля, як зазначено в посиланні [10]. Вчені підкреслили, що підхід не в змозі визначити оптимальний шлях і представляє проблеми в його реалізації.

В [12] представлено використання онлайн-планування шляху робота

(HIRO), яке може значно прискорити виявлення шляхів без зіткнень у порівнянні зі звичайними методами, незалежно від того, чи має робот попередні знання про навколишнє середовище. Для швидкого пошуку шляху HIRO використовує як інформовану евристику, так і детерміновану дорожню карту. Імовірнісна дорожня карта (PRM) широко використовується в навігації мобільних роботів через її простоту. Коли в навколишньому середовищі є вузькі проходи, ефективність PRM значно знижується.

В [13] автори запропонували вдосконалений алгоритм імовірнісної дорожньої карти на основі потенційних полів. Проблема планування шляху може бути успішно вирішена за допомогою імовірнісної дорожньої карти (PRM) в умовах із численними складними обмеженнями та великими розмірами. Його недоліки включають низьку якість та ефективність планування шляху в обмеженому просторі та динамічному середовищі.

Автори [14] запропонували використовувати динамічне PRM-змішане потенційне поле для планування шляхів для мобільного робота в середовищі з більш ніж одним виміром.

### 1.2.2 Розкладання комірки (CD)

Щоб визначити маршрут, що з'єднує початкову та цільову конфігурації, безперешкодний конфігураційний простір поділяється на набір відсіків.

Як показано на рис. 1.6 [14], встановлення зв'язку між початковою та кінцевою комірками та подальше встановлення з'єднання через серію проміжних комірок визначає клітинні зв'язки. Автори [15] і [16] запропонували декомпозицію з використанням квадродерева.

Великі осередки сітки розділяють середовище; однак, коли клітинка сітки заповнена лише частково, вона розбивається на чотири менші частини однакового розміру, поки не стане порожньою.

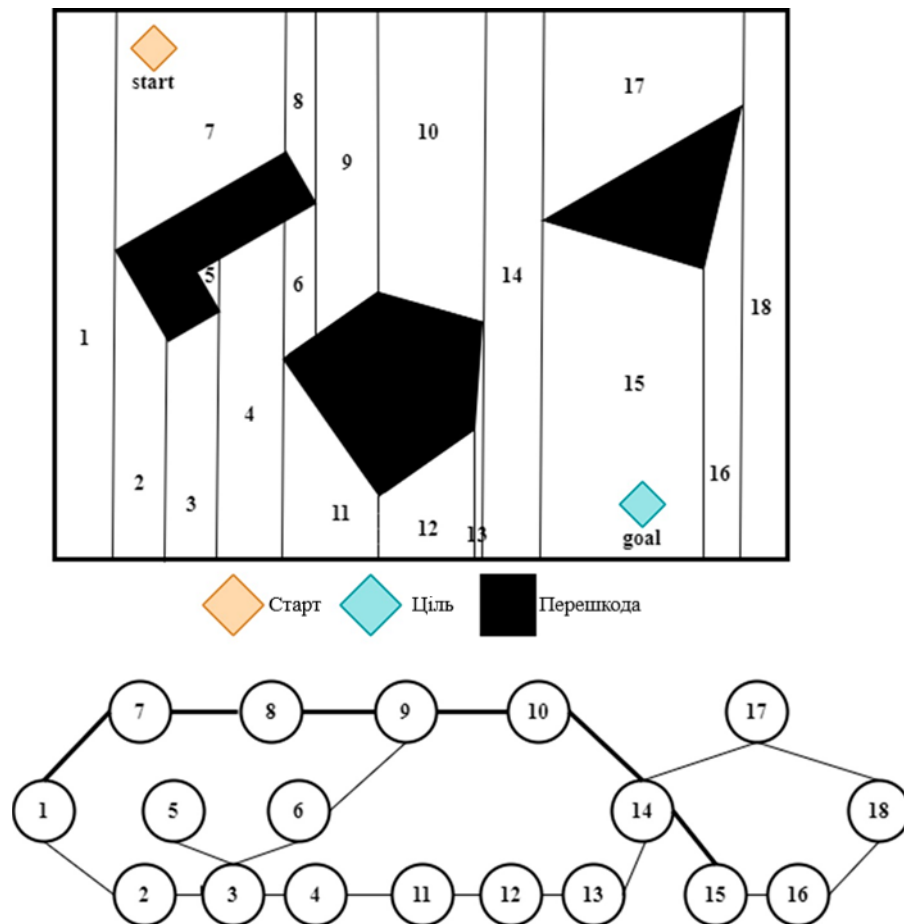


Рисунок 1.6 – Приклад точного розкладання клітини

Недоліком цієї системи є її нездатність оновлювати програму, коли надходить нова інформація (наприклад, положення перешкоди), що робить її неефективною в динамічних середовищах. Автор [17] довів, що існує проблема статичної конфігурації великої розмірності з плануванням шляху. Він знайшов рішення проблеми планування шляху для роботизованих платформ, які нагадували ланцюги та лабіринт. Використовуючи дані CAD, автори [18] представили стратегію планування шляху для мобільних роботів.

### 1.2.3 Штучні потенційні поля (APF)

Нинішній підхід включає простір конфігурації, який складається з двох різних сил, а саме сили відштовхування, яка діє таким чином, щоб відштовхнути робота з перешкод, як показано на рис. 1.7, і сили тяжіння, яка притягує робота до бажаної конфігурації мети. Ця гіпотетична сила, показана

на рис. 1.7, штовхає робота в потрібному напрямку та не дає йому рухатися до бар'єру. Тут робот рухається вздовж негативного градієнта, щоб обійти перешкоду та дістатися до потрібного місця.

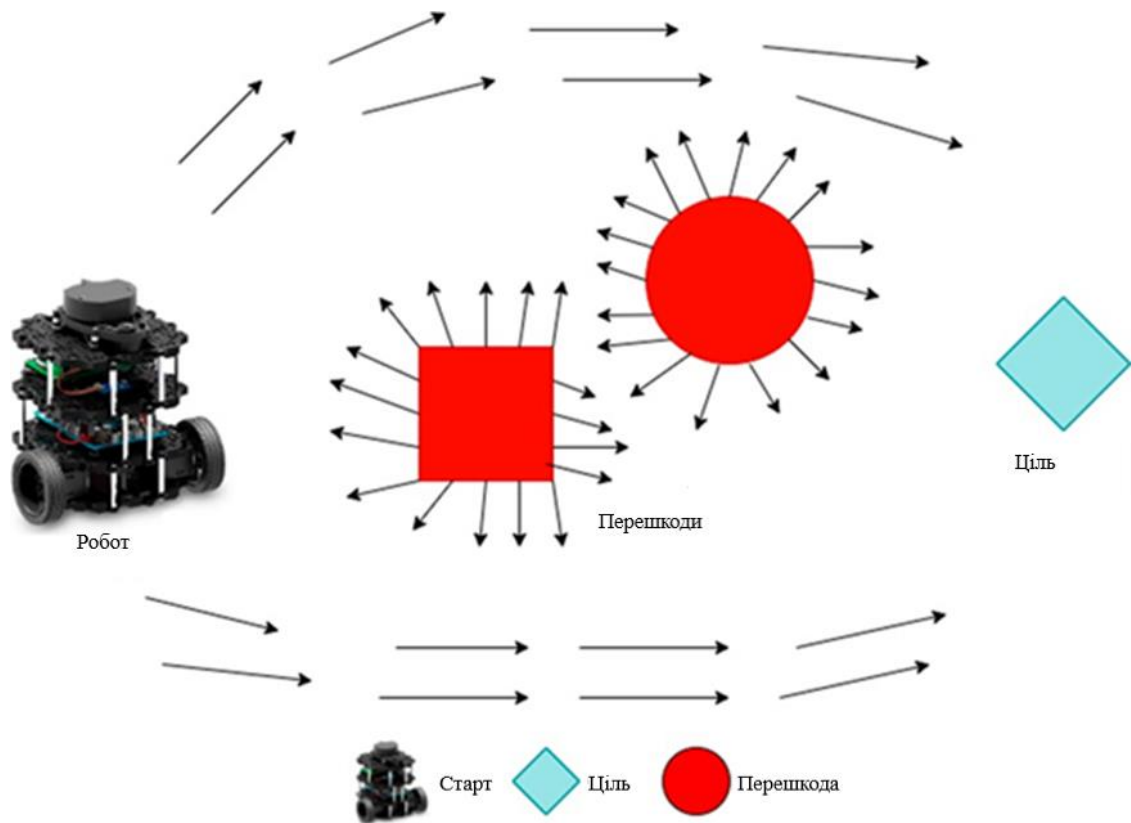


Рисунок 1.7 – Функції притягування та відштовхування

В [19-21] представили застосування цього методу для навігації мобільного робота. В [22-23] обговорили нову техніку уникнення перешкод у невивченому середовищі з використанням APF. Щоб обійти проблему локального мінімуму, вони використали гармонічну функцію.

Навігаційні стратегії розділені на дві категорії після обширного аналізу наукових статей, цитованих у літературі: евристичні підходи та класичні підходи. До кількох десятиліть тому переважна більшість досліджень у сфері робототехніки проводилася виключно традиційними методами. Традиційні методи демонструють кілька обмежень, таких як їх висока обчислювальна складність, сприйнятливність до захоплення локальних мінімумів, нездатність керувати максимальною невизначеністю, залежність від точних даних про

навколишнє середовище, необхідність точного сенсорного механізму для навігації в реальному часі та різноманітні додаткові недоліки.

Як наслідок, при використанні традиційного підходу ніколи не можна сказати, чи буде знайдено рішення, чи найкраще припустити, що воно не буде знайдено. Ці методи ненадійні та непередбачувані, що робить їх крихкими при використанні в режимі реального часу.

Незважаючи на зусилля кількох дослідників визначити недоліки традиційних методологій і розробити інноваційні методи, евристичні підходи продовжують перевершувати ці методи, включаючи APF і певні гібридні алгоритми, у сценаріях реального світу.

### 1.3 Аналіз технічного завдання

Додаток є інтегрованою та повністю автономною системою для виконання наскрізних завдань з вибору та розміщення у невідомих статичних середовищах.

Мобільний маніпулятор оснащений датчиком Kinect, встановленим на базі (надає карту глибини сканування області) і датчиком лінійного сканування Нокуо в руці (встановлений на зап'ясті руки, причому останній суглоб використовується для отримання карти глибини сканування області) і використовує ці датчики для дослідження навколишнього середовища.

Ключовим аспектом інтегрованої системи є те, що планувальник працює в тандемі з модулями дослідження бази та рук (перегляд планування), які досліджують невідоме середовище.

На відміну від інших реалізованих планувальників мобільних маніпуляторів, припускаємо невідомі ділянки середовища як перешкоди, а не вільні, і область повинна бути просканована вільно, перш ніж робот переміститься туди.

Інтегрована та автономна система має переваги:

а) відмінність від існуючих інтегрованих систем планування полягає в

тому, що вона досліджує невідоме середовище, розглядаючи невідомі регіони як перешкоди, вибирає об'єкт (як тільки об'єкт вважається таким, що знаходиться у відомому регіоні), а потім далі досліджує навколишнє середовище з об'єктом у руках і розміщує його в цільовому місці лише після того, як місце розташування є вважається таким, що знаходиться у відомому вільному регіоні;

б) він поєднує дві різні схеми дослідження в одну – використовує техніку дослідження на основі кордону для бази та техніку дослідження на основі максимізації приросту інформації (у робочому просторі) для руки;

в) він інтегрує сканування з кількох датчиків, а потім використовує їх для планування бази та огляду руки.

## 2 АЛГОРИТМІЧНЕ ТА АПАРАТНЕ ЗАБЕЗПЕЧЕННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ

Враховуючи положення кінцевого ефектора вибору та місця, завдання автономного мобільного маніпулятора полягає в тому, щоб досліджувати навколишнє середовище, вибирати об'єкт, як тільки він опиняється у відомому регіоні, а потім досліджувати решту середовища (якщо потрібно) з об'єктом у руках, щоб розмістити його в цільовому місці (позицію місця). Ми припускаємо, що на початку відомо про дуже маленьку область навколо мобільного маніпулятора, але крім цього невідоме все оточення. У майбутньому хват також буде вирішуватися системою, а поки ми надаємо позицію захоплення.

### 2.1 Компоненти системи

Модель мобільного маніпулятора SFU, як у симуляції, так і у фізичному середовищі, складається з мобільної бази powerbot, 6 DOF schunk powercube arm, встановленого на основі, 2-пальцевого захвату schunk, 2D-датчика дальності LMS100, розміщеного перед мобільною базою (використовується для SLAM), датчика глибини та зображення Kinect 3D, встановленого на основі, та легкого датчика дальності hokuyo 2D, встановленого на захваті (щоб діяти як око в руці), як показано на рисунку 2.1.

Використання датчиків:

1) датчик «око в руці» (у нашому випадку Hокуюо) краще досліджує області навколишнього середовища, які інакше закриті для Kinect, встановленого на базі. Оскільки датчик «око в руці» може забезпечити лише сканування ліній, тому в NBV-A (наступний найкращий вигляд руки) датчик обертається, щоб зробити сканування області, збираючи всі скани ліній під час обертання;

2) другий датчик (Kinect) доданий для онлайн-моніторингу виконання шляху, однак, опинившись там, він також діє як додатковий датчик для дослідження навколишнього середовища. Це потрібно ще й тому, що Хокуйо погано працює з чорними поверхнями та більшістю поверхонь в реальному середовищі (RAMP Lab) чорні. Kinect як датчик «око в руці» (замість Хокуйо для прискорення дослідження) не може визначати відстань до 1 метра (біля кліппінгу) і, отже, не може сканувати сусідні регіони. Третій датчик, LMS 100, встановлений у передній нижній частині бази, використовується для локалізації мобільної бази.

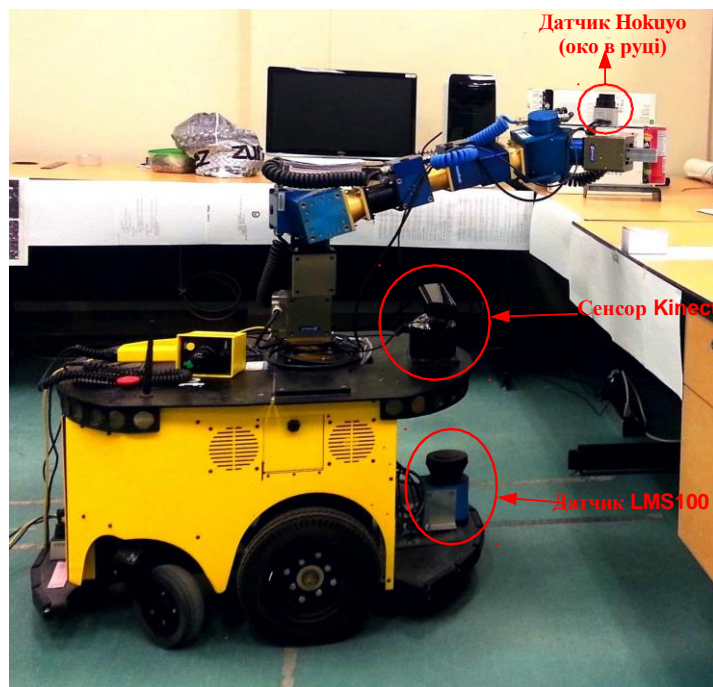


Рисунок 2.1 – Мобільний маніпулятор СФУ і місця кріплення різних датчиків [14]

## 2.2 Опис архітектури системи

Архітектура нашої інтегрованої та автономної системи описана на рисунку 2.2. Єдиними вхідними даними для системи є положення вибору та розміщення кінцевого ефектора. Для кожної заданої позиції обчислюються відповідні можливі базові позиції та конфігурації маніпулятора. Зауважте, що

на поточному етапі статус колізії цих базових поз і конфігурацій маніпулятора не може бути перевірений, оскільки оточення невідоме. Метод обчислення дійсної базової позиції та конфігурації маніпулятора, що відповідає позиції кінцевого ефектора.

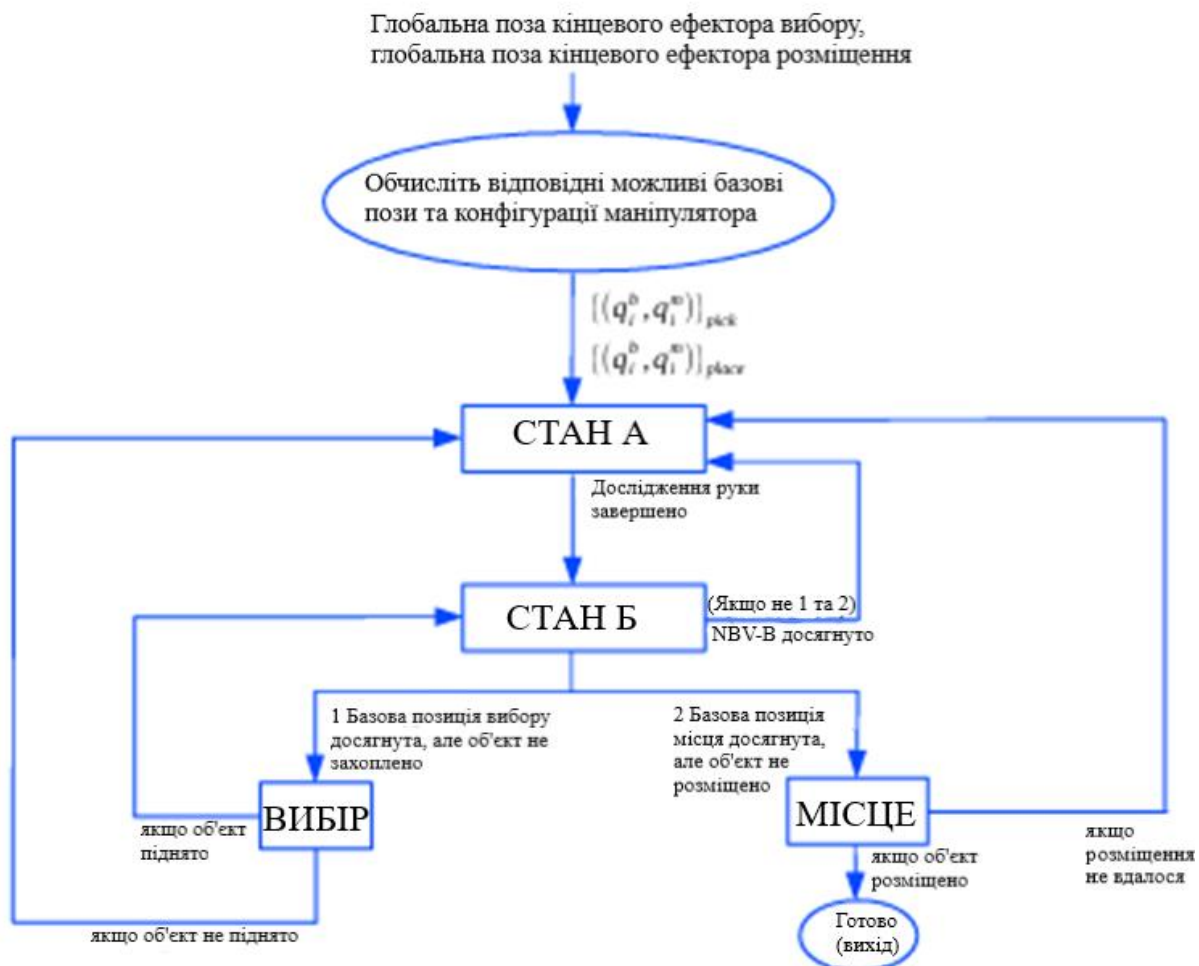


Рисунок 2.2 – Інтегрована та автономна система для виконання завдань у невідомих умовах

Спочатку випадковим чином вибирається базова позиція з області диска (обмежена досяжністю руки), а потім задана позиція кінцевого ефектора (медіатор або місце) перевіряється на досяжність з позиції вибірки основи за допомогою зворотної кінематики.

У будь-який момент часу система може перебувати в одному з чотирьох станів: СТАН А, СТАН Б, ВИБІР і МІСЦЕ. На початку система запускається

в СТАН А. Цей модуль використовує датчик «око в руці» для дослідження локальної області навколо мобільного маніпулятора. Як тільки локальний регіон повністю вивчений, система змінює свій стан на СТАН Б. Загалом, одним із завдань цього модуля є перевірка різноманітності базових позицій вибору та розміщення.

Якщо будь-яка з них досяжна, то планується шлях для переміщення мобільного маніпулятора в цю базову позицію. Залежно від успіху, стан змінюється на ВИБІР або МІСЦЕ. Якщо жодна з цих базових позицій вибору або місця не вважається досяжною, то модуль СТАН Б досліджує навколишнє середовище, досягаючи результату СТАНУ Б, а потім стан змінюється на СТАН А. Якщо в якийсь момент часу стан змінюється на ВИБІР, тобто досягається позиція бази вибору, то модуль ВИБІР планує вибір позиції кінцевого ефектора та захоплення об'єкта.

Якщо захоплення проходить успішно, то стан перемикається назад на СТАН Б, щоб дослідити середовище, що залишилося, для завершення іншої частини мети, тобто для розміщення об'єкта. У міру того, як система досліджує все більше і більше навколишнього середовища (використовуючи СТАН Б і СТАН А), позиція бази місця буде доступною в певний момент часу. Як тільки базова позиція місця досягнута, то викликається модуль МІСЦЕ, щоб розмістити об'єкт у цільовому місці. Детальна інформація про ці модулі наведена з рис. 2.3 – рис 2.4.

СТАН А (рис. 2.3) створює локальну воксельну карту, дискретизоване сітчасте представлення 3D робочого простору у вигляді 3D растрового зображення (з вільним як 0, перешкодою як 1 і невідомим як 2), у фіксованій базовій позиції. Зауважимо, що межі Voxelmap повинні бути з досяжністю кінцевого ефектора. Статус (зайнято, вільно, невідомо) кожної воксельної комірки у Voxelmap оновлюється шляхом зв'язку з глобальним Octomap (octree представлення Voxelmap).

Потім обчислюються кордони (вільні клітинки поруч з невідомими). Якщо число кордонів нижче заданого порогу, то дослідження руки

припиняється, що стверджує, що місцева область навколо мобільного маніпулятора повністю вивчена. Якщо ні, то для обчислення набору позицій кінцевого ефектора, розташованих у порядку пріоритетності отримання інформації (що є кількістю невідомих вокселів у полі зору датчика в запланованій позиції).

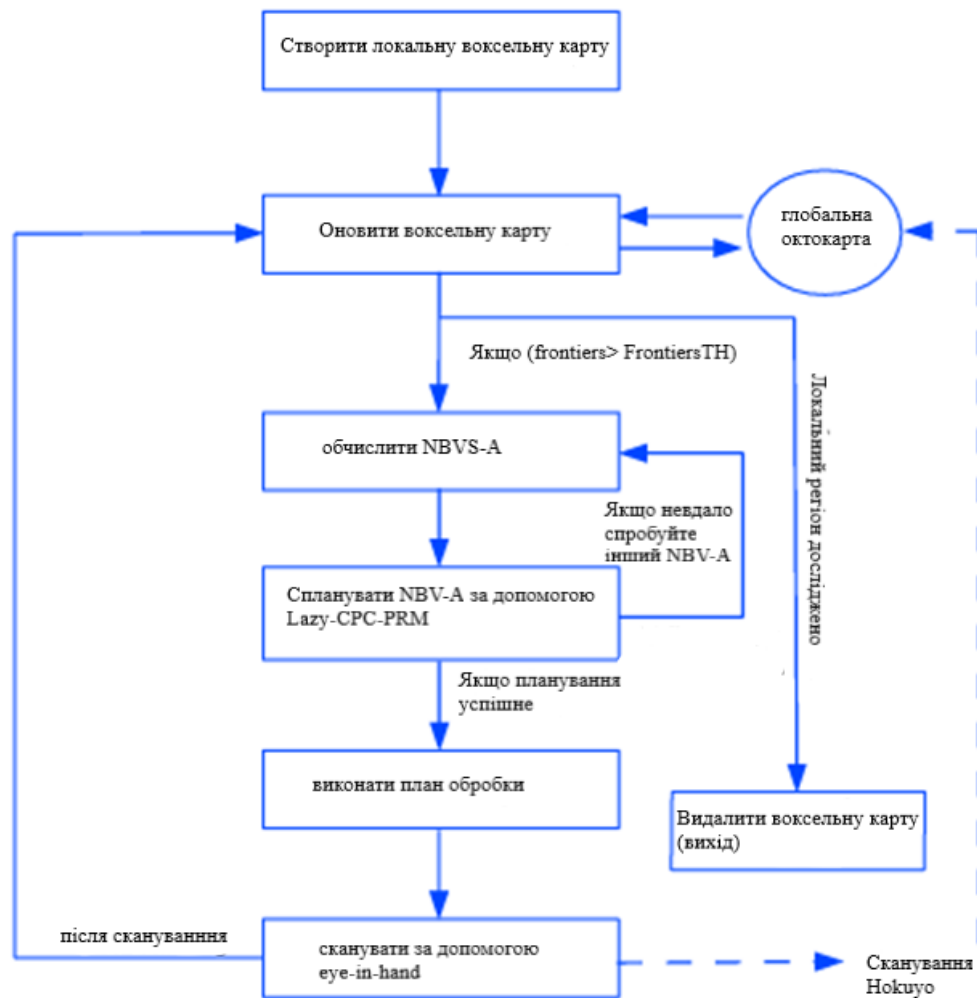


Рисунок 2.3 – Модуль системи (СТАН А)

### 2.3 Опис модулів

Процедура обчислення виглядає наступним чином: зразки планування перегляду руки дійсні (ІК існує, і рішення також без зіткнень з об'єднаними межами) позує кінцевий ефектор і обчислює виграш інформації в кожній позиції датчика, моделюючи модель датчика, а потім повертає набір цих

позицій у порядку високого приросту інформації у верхній частині. Потім кінцевий ефектор переміщується в положення next-best view (NBV-A) для проведення сканування за допомогою датчика Хокуйо «око в руці».

- Зверніть увагу, що скани вставляються в глобальний Octomap, а не локальний Voxelmap. Lazy-CPC-PRM використовується для планування шляху маніпулятора для ІК рішення NBV-A як мети. Після сканування локальний Voxelmap оновлюється, і процедура триває до тих пір, поки Voxelmap не буде досліджений або не буде досягнуто максимальної кількості ітерацій.

СТАН Б (рис. 2.4) спочатку перевіряє, з відомою областю середовища, чи можливо перемістити мобільний маніпулятор для вибору базової позиції або розміщення базової позиції.

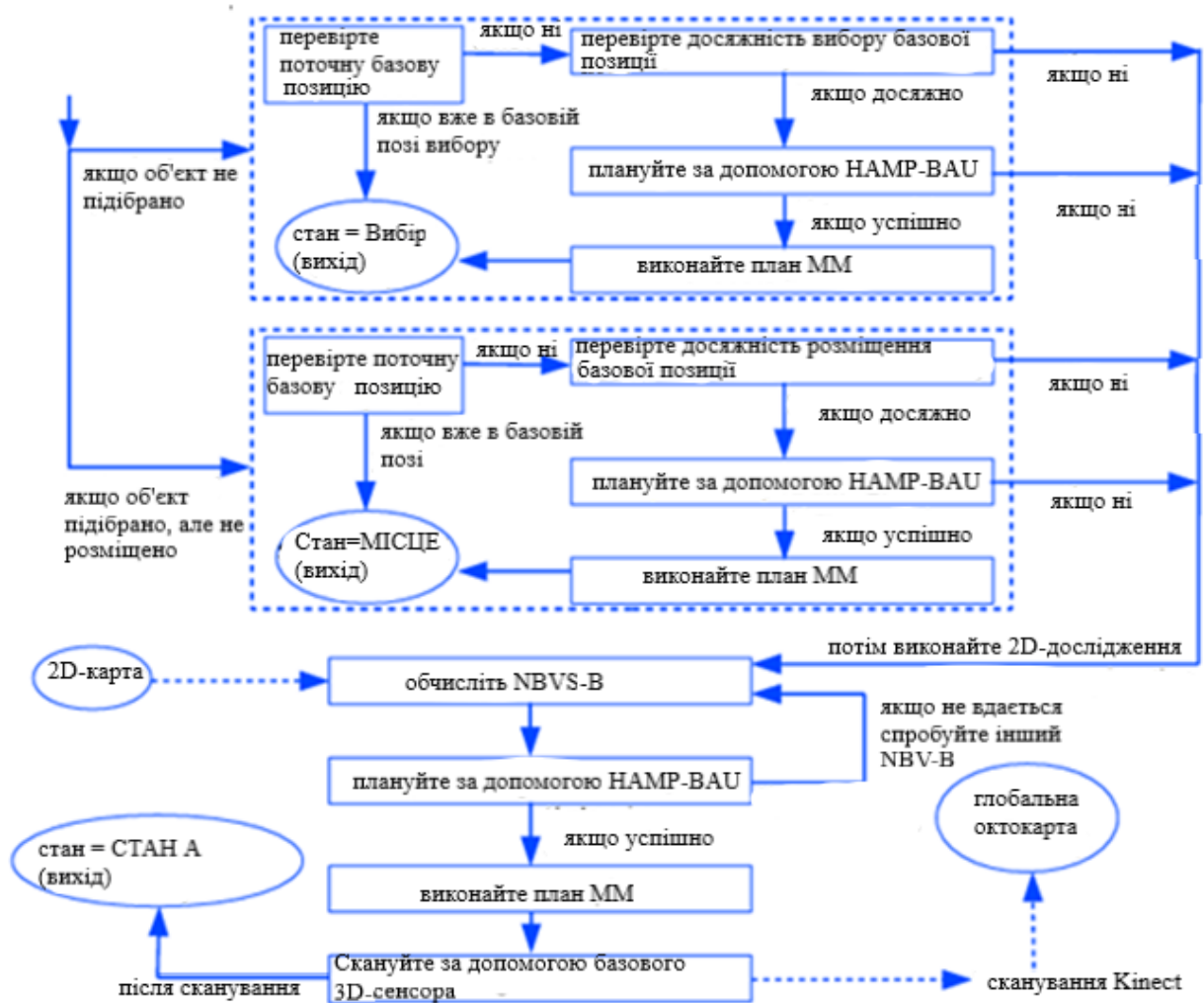


Рисунок 2.4 – Модуль системи (СТАН Б)

Якщо об'єкт ще не вибраний, то перевіряється стан колізії раніше обчислених базових позицій вибору. Якщо будь-яка з базових позицій виявляється вільною від зіткнень, то для планування шляху використовується НАМР-ВUA. Якщо досягнута позиція бази вибору (планування успішне) або мобільна база вже знаходиться в позиції бази вибору, то модуль СТАН Б просто перемикає стан на ВИБІР.

Аналогічно, якщо об'єкт захоплений, але не розміщений, перевіряється досяжність позиції бази без зіткнень, і в разі успіху модуль СТАН Б переривається, перемикаючи стан на МІСЦЕ. Однак, якщо планування не вдається, або якщо вибрати і розмістити базові позиції не є вільними від зіткнення з досліджуваною областю, то база наступна найкращий вид (NBV-B) досягається для дослідження навколишнього середовища за допомогою датчика (Kinect), встановленого на основі. Знову ж таки, скани датчиків вставляються в глобальну карту Octomap.

Після сканування стан перемикається на СТАН А, щоб досліджувати місцевий регіон, цього разу з іншої базової позиції. Планування базового вигляду використовує 2D сітку зайнятості (дискретизоване сітчасте представлення 2D робочого простору у вигляді растрового зображення з вільним як 0, перешкодою як 1 і невідомим як 2) для обчислення NBV-B, і ця карта складається з серії кроків, як показано на рисунку 2.5.

Сканування сенсорів та представлення світу



Рисунок 2.5 – Сканування датчиків і зображення світу (2D і 3D)

Модулі ВИБІР і МІСЦЕ, які є досить стандартними для захоплення домену. Модуль ВИБІР викликається, як тільки мобільний маніпулятор досягає базової позиції, звідки можна захопити об'єкт. Раніше обчислена інформація (базові позиції та конфігурації маніпулятора) може бути некорисною для вибору кінцевої ефекторної позиції (позиція захоплення) більше не може бути корисною, оскільки через невизначеність у положенні мобільної бази, мобільний маніпулятор (в основному мобільна база) не точно досягає передбачуваної позиції бази вибору. Таким чином, необхідно обчислити або нову конфігурацію маніпулятора з досягнутої базової позиції, або нову допустиму позиції захоплення (і, отже, відповідну досягнути конфігурацію маніпулятора), з перебуванням у безпосередній близькості від вже заданої позиції захоплення. У випадку останнього (регулювання захоплення) ми використовуємо простий підхід, який полягає в наступному: позиція захоплення позначається як  $p = [x, y, z, \alpha, \beta, \gamma]$ , позиція кінцево-ефекторної рамки (розташована в центрі щелеп захвату) по відношенню до нерухомої рами, наприклад, базової рами маніпулятора.

У прикладах симуляції та реального експерименту нова допустима позиція захоплення обчислюється шляхом зміни  $\gamma$  при збереженні інших 5 параметрів незмінними. Як вже говорилося раніше, це дуже швидка і ситуативна спроба показати інтегровану систему. Для дійсної позиції захоплення (існує рішення ІК без зіткнень) ми також перевіряємо, чи є її позиції до та після захоплення дійсними, які зазвичай зміщені на 10 см (назад і вгору відповідно) від передбачуваної позиції захоплення.

## 2.4 3D картографування

Метод оцінки положення за допомогою 3D хмар точок навколишнього середовища складається з (1) 3D картографування з MMS та (2) локалізації мобільних роботів з використанням цих підготовлених карт.

На рисунку 2.6 показаний алгоритм локалізації потоку з фільтром

частинок за допомогою 3D карт оточення. Цей алгоритм вимагає (1) 3D-карт оточення на основі вимірювання MMS та (2) мобільного робота з конфігурацією датчика DR з енкодером, 1-осьовим гіроскопом та розпізнаванням навколишнього середовища за допомогою одного лазерного далекоміра (LRF).

Алгоритм виконується наступним чином:

1) векторами станів частинок є 2D положення  $x, y$  і орієнтація (кут рихання)  $\psi$ . Імовірнісні моделі мобільних роботів виготовляються з використанням енкодера на ведучому колесі мобільного робота та 1-осьового гіроскопа для ДР, обчислення положення та орієнтації та розподілу частинок;



Рисунок 2.6 – Блок-схема локалізації з використанням 3D хмар точок

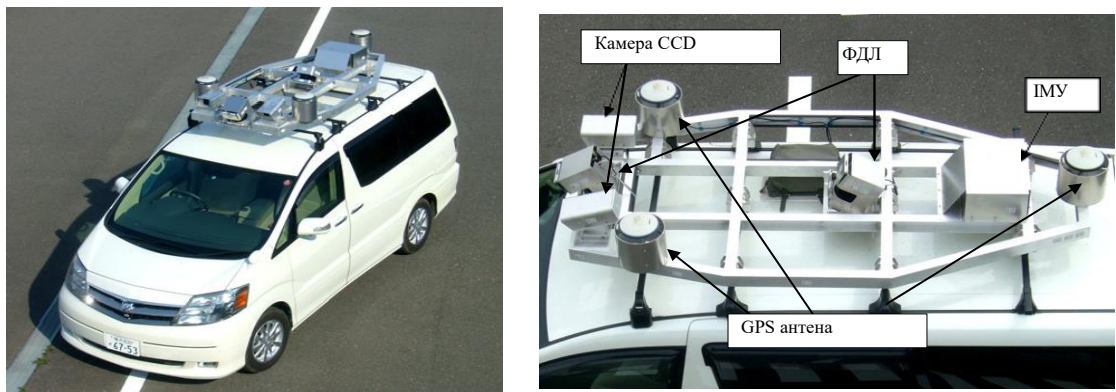
2) потім оцінюють висоту руху мобільного робота  $z$ , кут крену  $\theta$  та кут тангажу  $\varphi$  на основі перерозподілених частинок, припускаючи, що всі шини робота постійно контактують з дорожнім покриттям;

3) вимірювання LRF перетворюється в координати 3D-відображення на основі отриманого положення 6-DOF та положення для розрахунку ймовірності кожної частинки шляхом зіставлення з 3D-картами;

4) частинки повторно дискретизуються на основі отриманої таким чином

ймовірності, а середнє положення та положення повторно дискретизованих частинок беруться для позначення кінцевого положення та позиції мобільного робота.

На рисунку 2.7 показані конфігурації MMS і датчиків. MMS використовує D-GPS несучої фази та волоконно-оптичний гіроскоп (FOG) для вимірювання положення та положення MMS. Крім того, ми використовуємо GPS-гіроскоп/інерційну одиницю вимірювання (IMU) для об'єднання оцінок положення та положення; положення, оцінене за допомогою GPS-гіроскопа за допомогою трьох GPS антен, об'єднаних між собою за допомогою EKF [9].



а)

б)

а) MMS; б) конфігурація датчика

Рисунок 2.7 – Мобільна картографічна система

Два LRF (LMS291, SICK Inc.), що використовуються для 3D-вимірювань, дозволяють вимірювати відстань з зареєстрованою точністю  $\pm 35$  мм; Вони забезпечують поступеневе сканування в діапазоні  $180^\circ$ . Дані 3D-хмари точок навколишнього середовища збираються за допомогою синхронізації датчиків на основі інформації про час GPS і реконструюються з використанням положення та положення MMS та даних вимірювань, отриманих з LRF.

Ця система, яка дозволяє високоточно оцінювати положення та положення в міському або зеленому середовищі, або там, де супутникова передача, ймовірно, буде відключена, виявилася здатною проводити точні 3D-

вимірювання зовнішнього середовища в широкому діапазоні зі стандартним відхиленням ( $1\sigma$ ) 10 см [9].

На рис. 2.8 показана 3D-реконструкція сцени, яка використовує MMS для легкого отримання високоточних даних 3D-хмари точок для широкого спектру зовнішніх середовищ.

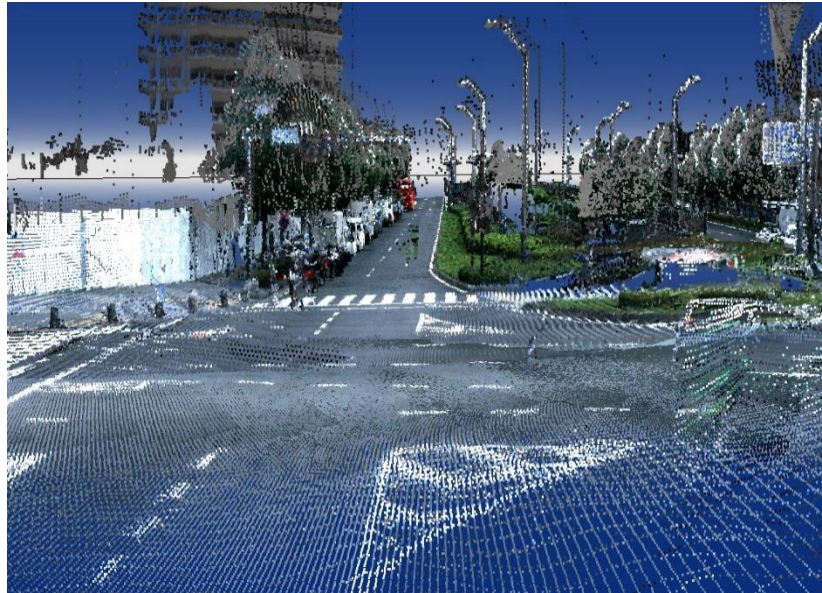


Рисунок 2.8 – Реконструкція 3D-сцени за допомогою MMS

Застосування таких даних 3D хмари точок для локалізації автономних мобільних роботів викликає такі питання:

1) комп'ютерам потрібен величезний обсяг пам'яті для обробки таких великих обсягів даних, а випадкова конфігурація даних у часовій послідовності негативно впливає на продуктивність пошуку;

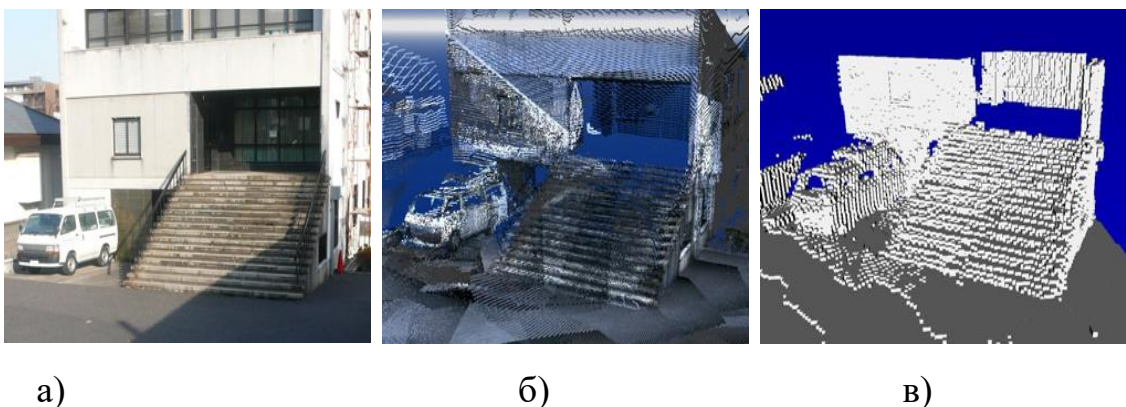
2) роздільна здатність 3D-вимірювання змінюється залежно від відстані вимірювання, що призводить до нерівномірності 3D-простору.

Щоб впоратися з цими проблемами та дозволити автономному мобільному роботу легко обробляти дані 3D-хмари точок, отримані за допомогою MMS для цілей локалізації, ми кількісно оцінили 3D-хмари точок у воксельному просторі для використання в 3D-картах середовища. Обсяг даних зменшується за рахунок рівномірного квантування та вилучення лише високоімовірнісних точок після того, як дані 3D-хмари точок обробляються

голосуванням у воксельний простір. У цих процесах, чим вища роздільна здатність воксельного простору, тим точніше відображаються 3D-форми. Однак для цього потрібен надзвичайно великий обсяг пам'яті. Тому ми кількісно розподілили дані в блоки  $100 \text{ мм} \times 100 \text{ мм} \times 10 \text{ мм}$  у напрямках осей  $x$ ,  $y$  та  $z$ . Ми використовували вищу роздільну здатність лише для напрямку осі  $z$ , щоб точно відтворити форму дорожнього покриття та дані про перешкоди. Такі 3D воксельні хмари точок представляють настільки розріджений 3D масив, що ми стиснули їх у розріджену матрицю для збереження в пам'яті комп'ютера.

На рисунку 2.9 показано воксельне визначення для створення 3D-карт з 3D хмар точок. Порівняння (рис. 2.9, а) фотографії навколишнього середовища і 3D хмари точок, отримані за допомогою MMS, показують, що середовище було точно представлено 3D-хмарами точок (рис. 2.9, б).

На рис. 2.9, в показана воксельна модель, яка підтверджує, що середовища були сконструйовані з неперервною роздільною здатністю. Для цих середовищ ми створили 3D воксельні карти для експериментальних середовищ  $100 \text{ мм} \times 100 \text{ мм}$ ; В середньому, навколо дорожнього покриття було включено чотири точки вимірювання. Стиснення 3D-воксельних карт у розріджену матрицю призвело до розміру 3 МБ у середовищі  $100 \text{ м} \times 100 \text{ м}$ .



а) фотографія навколишнього середовища; б) 3D-хмари точок;  
в) модель Voxel

Рисунок 2.9 – Воксельне визначення для створення 3D-карт

## 3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ АВТОМАТИЗАЦІЇ ПОШУКУ ШЛЯХУ РОБОТИЗОВАНОЇ ПЛАТФОРМИ

Для вирішення проблеми планування шляху інтелектуальних транспортних засобів у невідомих умовах ми розробили метод побудови карт і траєкторій для мобільних роботів на основі злиття мультисенсорної інформації. Основні нововведення полягають у наступному:

1) в умовах невідомого середовища пропонується метод позиціонування та картографування в реальному часі, заснований на злитті LiDAR, камери глибини та інерційного блоку вимірювання;

2) запропоновано метод планування гібридного шляху на основі комбінації вдосконаленого алгоритму колонії мурах та методу динамічного вікна. За допомогою цього методу можна оптимізувати кут повороту шляху, уникнути зіткнення з перешкодами, підвищити ефективність планування шляху, а також реалізувати механізм автоматичного зворотного зв'язку мобільного робота.

### 3.1 SLAM на основі мультисенсорного злиття

У галузі навігації роботів проблему SLAM можна просто описати так: робот, що рухається в невідомому середовищі, може оцінювати власну позицію відповідно до інформації про позицію та карту і може будувати інкрементальну карту в процесі руху для досягнення автономного обходу перешкод та навігації [22]. Для опису задачі SLAM рух транспортного засобу за безперервний проміжок часу розкладається на дискретні моменти  $t = 1, 2, \dots, k$ , і відповідні позиції моменту  $x_1, x_2, \dots, x_k$  - траєкторія руху транспортного засобу. Для нелінійної системи в реальному процесі руху модель руху і модель спостереження системи можуть бути виражені

наступним чином:

$$\begin{cases} x_k = f(x_{k-1}, u_k, w_k) \\ z_k = h(x_k, v_k) \end{cases} \quad (3.1)$$

де  $x_k$  – представляє позицію робота;

$z_k$  – представляє спостережуване значення системи;

$f(\cdot)$  – функція рівняння руху системи;

$h(\cdot)$  – функція рівняння системного спостереження;

$U_k$  – керуюче значення системи;

$w_k$  – шум системного процесу;

$v_k$  – шум спостереження системи.

### 3.2 Опис алгоритму

На основі 2D LiDAR та датчиків глибинного бачення пропонується структуру SLAM для автономної локалізації та картографування транспортних засобів на основі злиття кількох датчиків, щоб компенсувати проблему, пов'язану з неможливістю сприйняття низьких перешкод у навколишньому середовищі. Алгоритм реалізує інформаційне злиття 2D LiDAR і датчика глибинного бачення, покращує здатність автомобіля сприймати навколишнє середовище, підвищує здатність автомобіля виявляти перешкоди в складному середовищі, дозволяє уникнути недоліку низької точності виявлення через продуктивність одного датчика, а також покращує точність і стабільність SLAM мобільного робота.

Загальна структура алгоритму показана на рисунку 3.1.

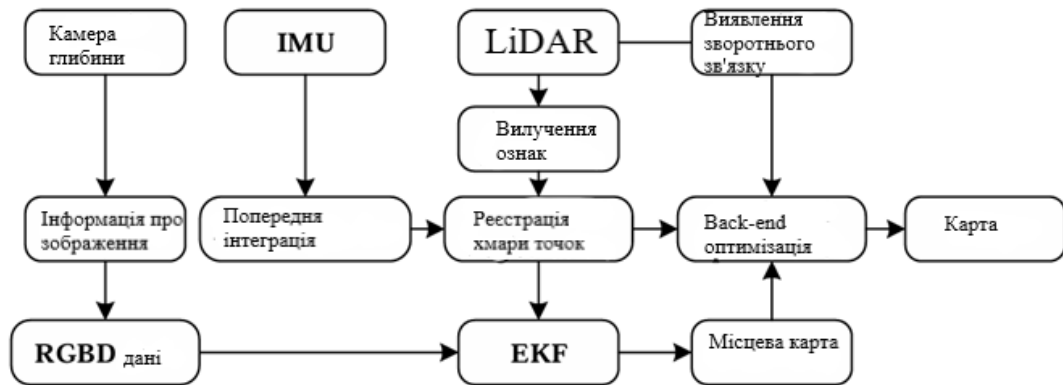


Рисунок 3.1 – Загальна структура алгоритму SLAM

Загальний процес складання алгоритму в основному поділяється на такі чотири частини:

1) калібрування датчика: камера, LiDAR та IMU калібруються в часі та просторі, і тому інформація, що виводиться датчиком, знаходиться в одній площині та часі;

2) попередня обробка даних: обробка даних лазерної хмари точок та інформації з камери. IMU попередньо інтегрований для отримання відносної матриці трансформації позиції транспортного засобу, що забезпечує основу трансформації для узгодження рамки хмари точок LiDAR;

3) злиття даних EKF: розширений фільтр Калмана використовується для об'єднання даних виявлення лазера та даних візуального виявлення для створення локальної карти;

4) побудова карти та виявлення петлі: позиція робота в реальному часі використовується як центр кола,  $R$  – радіус для окреслення кола виявлення, а алгоритм ICP лазерної хмари точок використовується для виконання виявлення зворотного зациклення на траєкторії робота.

### 3.3 Планування шляху для мобільних роботів

3.3.1 Глобальне планування шляху на основі вдосконаленого алгоритму мурашиної колонії

Запропоновано два вдосконалення для традиційних алгоритмів з точки зору повільної швидкості збіжності та задач локальної оптимізації:

1) використання вдосконаленої евристичної функції, щоб забезпечити початковий вибір колонії мурах для вказівки напрямку;

2) використання гібридної стратегії для оновлення феромону.

Початковий стан: у початковому стані ймовірність того, що мурахи виберуть можливий шлях, дорівнює, і мураха визначить напрямок наступного вузла відповідно до вмісту феромонів у точці переходу стану на шляху вибору. Встановлення ймовірності переходу стану  $ant\ b$  з точки  $I$  в точку  $g$  на графіку сітки як  $p^{(b)}(t)$ , вміст феромонів  $T_{ig}(t)$  з точки  $i$  в точку  $g$ , евристична функція  $N_{ig}(t)$ , Визначаються феромонний евристичний фактор  $A$  та очікуваний евристичний коефіцієнт  $W$ . Тоді формула ймовірності переходу стану має такий вигляд:

$$p_{ig}^{(b)}(t) = \begin{cases} T_{ig}^a \cdot N_{ig}^w(t) / \sum_{g \in all(i)} T_{ig}^a \cdot N_{ig}^w(t) \\ 0, otherwise \end{cases} \quad (3.2)$$

де всі  $(i)$  являють собою множину вузлів передачі, які в  $b$  може вибрати в вузлі  $I$ , а  $N_{ig}(t)$  обернено пропорційна відстані  $l_{ig}$  між вузлами  $i$  до  $g$ .

Поліпшена евристична функція: На основі квадрата суми відстані від поточного вузла до наступного вузла і відстані від наступного вузла до мети в [23] вводиться евристична функція для прискорення швидкості пошуку мурашиної колонії, яка становить:

$$N_{ig}(t) = 1 / (\sigma \cdot l_{ig} + (1 - \sigma) \cdot l_{iE})^2 \quad (3.3)$$

де  $\sigma \in [0, 1]$ , що визначається середовищем реального часу;  $l_{iE}$  представляє відстань між поточним вузлом і наступним вузлом; а  $l_{iE}$  представляє відстань між поточним вузлом і цільовою точкою, і збільшує вибір колонії мурах і

ефект наведення на наступну цільову точку, що допомагає скоротити час пошуку мурашиної колонії.

Оновлення феромонів: Для того, щоб прискорити час пошуку мурах, зменшити ймовірність вибору пройденого маршруту, а також уникнути занадто низької концентрації феромонів або накопичення після випаровування, необхідно визначити максимальне значення  $T_{max}$  і мінімальне значення  $T_{min}$  концентрації значень між двома точками.

У даній роботі були визначені максимальні і мінімальні значення феромонів в залежності від ступеня випаровування і оптимального шляху феромону, який становить

$$\begin{cases} T_{max} = [1/2(1 - \lambda)] \times [1/D_s] + 1/D_s \\ T_{min} = T_{max}/c \end{cases} \quad (3.4)$$

де  $D_s$  – оптимальна довжина шляху після певної ітерації;  
 $c$  – кількість циклів кожного разу.

Після того, як мурахи закінчують кожен цикл, інформація про феромони на шляху змінюється. Для того щоб отримати оптимальний шлях, феромон на всьому шляху потрібно відрегулювати.

Умови оновлення наведені так, як показано в рівнянні (3.5). У той же час, щоб запобігти використанню колонією мурах феромону на місцевому оптимальному шляху для планування шляху, збільшення концентрації феромонів  $b$ -го мурашки між двома точками приймає стратегію оновлення, показану в (3.6), щоб гарантувати, що кожна мураха може знайти можливий шлях.

$$T_{ig}(t + 1) = (1 - \lambda) \cdot T_{ig}(t) + \Delta T_{ig}(t) \quad (3.5)$$

$$\Delta T_{ig}(t) = \sum_{b=1}^m \Delta T_{ig}^b(t) \quad (3.6)$$

$$\Delta T_{ig}^b = \begin{cases} T_{\min} & \Delta T_{ig}^b + (1 - \lambda)T_{ig}^{b-1} < T_{\min} \\ 1/r_s & T_{\min} < \Delta T_{ig}^b + (1 - \lambda)T_{ig}^{b-1} < T_{\max} \\ T_{\max} & \Delta T_{ig}^b + (1 - \lambda)T_{ig}^{b-1} > T_{\max} \end{cases} \quad (3.7)$$

де  $\lambda$  – коефіцієнт випаровування феромонів,  $\lambda \in [0, 1]$ ;

$\Delta T_{ig}(t)$  – являє собою приріст феромонів між вузлами  $i$  і  $g$ ;

$m$  – являє собою приріст феромонів  $b$ -го мурашки між вузлами  $i$  і  $g$ ,  $m \in [23]$ ;

$r_s$  – оптимальне рішення, знайдене на сьогоднішній день.

### 3.3.2 Планування місцевих доріжок

Глобальне планування шляху може спланувати глобальне оптимальне рішення маршруту, але для цього необхідно заздалегідь знати точну інформацію про навколишнє середовище. Інтелектуальна платформа робота інтегрує інформацію про перешкоди в карту навколишнього середовища в режимі реального часу за допомогою власних датчиків навколишнього середовища, коли середовище водіння інтелектуального робота змінюється, наприклад, коли з'являється невідома перешкода. Потім траєкторія руху інтелектуальної платформи робота перепланується за допомогою методу планування місцевого шляху. Це робиться для того, щоб уникнути перешкод. Після завершення уникнення локальних перешкод інтелектуальна платформа робота продовжує рухатися відповідно до глобального оптимального рішення.

Алгоритм динамічного вікна може реалізувати планування пробного шляху та уникнення перешкод у режимі реального часу. У процесі роботи розумного автомобіля немає груп просторів швидкостей  $(vt, \omega t)$ , і допустима траєкторія руху робота може бути змодельована в межах віконної області часу  $t$ . При цьому допустима траєкторія оцінюється за деякими метриками, щоб вибрати швидкість, що відповідає оптимальній траєкторії руху до мобільної платформи. З огляду на існування декількох можливих

змодельованих траєкторій руху в просторі швидкостей, в даній роботі розрахована функція оцінки в поєднанні з інформацією про глобальне планування шляху, щоб гарантувати, що остаточне планування локального шляху є глобальним оптимальним шляхом. Розроблена функція оцінювання полягає в наступному:

$$G(v, \omega) = \mu \cdot goal(v, \omega) + \beta \cdot d(v, \omega) + d(v_m, \omega_m) + \lambda v_e(v, \omega) \quad (3.8)$$

де  $goal(v, \omega)$  – кут між вектором, що вказує на мету, і вектором, що з'єднує початкову точку і поточне положення;

$d(v, \omega)$  – відстань по вертикалі між статичною перешкодою і поточною траєкторією, що вказує на ціль;

$d(v_m, \omega_m)$  – відстань по вертикалі між невідомою перешкодою і поточною траєкторією наведення цілі;

$v_e(v, \omega)$  – функція оцінки поточної швидкості, що вказує на цільову траєкторію;

$\mu, \beta, \varphi$  і  $\lambda$  – ваговими коефіцієнтами.

### 3.3.3 Планування шляху в поєднанні з покращеним алгоритмом колонії мурах та алгоритмом динамічного вікна

Блок-схема алгоритму показана на рисунку 3.2. Загальний опис блоків:

- використовуйте бортові датчики для отримання інформації про навколишнє середовище, щоб зрозуміти власне позиціонування автомобіля та побудувати растрову карту;

- використовуйте вдосконалений алгоритм колонії мурашок, щоб завершити глобальне планування шляху розумного транспортного засобу та отримати оптимальний шлях;

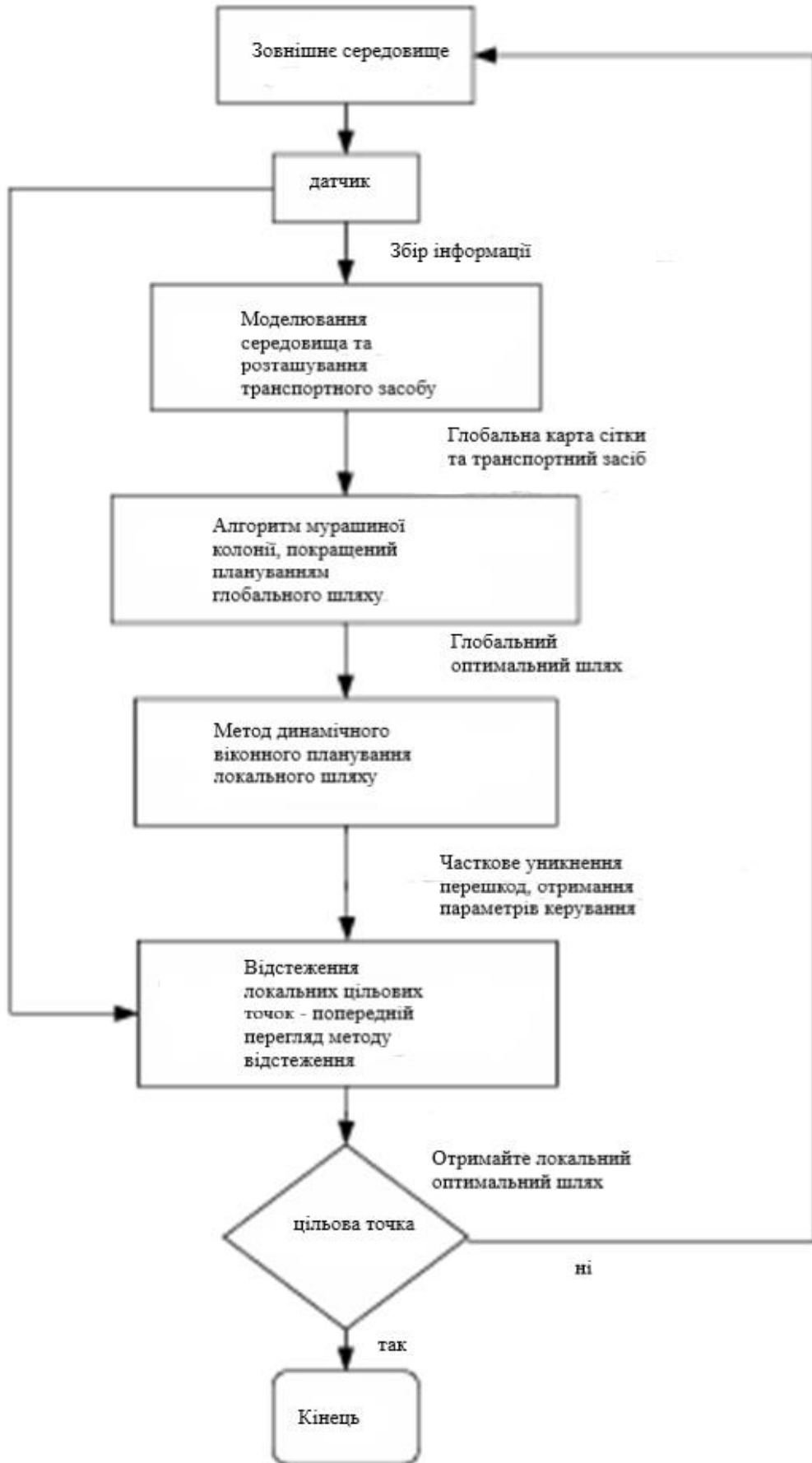


Рисунок 3.2 – Алгоритм злиття

- відповідно до нової інформації про перешкоди, отриманої датчиком автомобіля в режимі реального часу, інтегруйте інформацію в сітку карти та використовуйте метод динамічного вікна, щоб завершити локальне уникнення перешкод;

- поєднайте глобальний шлях планування, плануйте локальні рухомі цільові точки в режимі реального часу та використовуйте метод попереднього відстеження для відстеження цільових точок відповідно до контрольних параметрів у процесі планування шляху, щоб досягти уникнення перешкод у режимі реального часу та отримати місцевий оптимальний шлях;

- визначте, чи є місцева рухома цільова точка кінцевою цільовою точкою; якщо ні, перейдіть до кроку 1. Якщо так, інтелектуальний транспортний засіб досягає цільової точки, і алгоритм закінчується.

### 3.4 Практична реалізація

Для роботи була обрана мова програмування Python. Для створення програми використовувались наступні бібліотеки:

- ContourPy – це бібліотека Python для розрахунку контурів двовимірних чотирикутних сіток;

- fontTools – бібліотека для роботи зі шрифтами, написана на Python;

- Matplotlib – популярна Python-бібліотека для візуалізації даних. Вона використовується для створення будь-яких видів графіків: лінійних, кругових діаграм, рядкових гістограм та інших – залежно від завдань;

- NumPy – це бібліотека Python, яку застосовують для математичних обчислень: починаючи з базових функцій і закінчуючи лінійною алгеброю;

- Pillow – найпопулярніша бібліотека для роботи із зображеннями в Python. За допомогою неї зображення можна відкривати, обертати, накладати фільтри і навіть працювати з окремими пікселями;

- Python-dateutil – це бібліотека, яка надає додаткові можливості для роботи з датами та часом у Python. Вона розширює функціонал стандартного

модуля `datetime`, додаючи нові методи та функції для роботи з датами, часом та інтервалами;

- `six` – це бібліотека Python з відкритим вихідним кодом та набір інструментів для вилучення даних із PDF-документів;

- `Tkinter` – бібліотека призначена для організації діалогів у програмі за допомогою віконного графічного інтерфейсу;

- `Auto Py to Exe` – це утиліта що спрощує перетворення скриптів Python у виконувани файли (`.exe`). Вона основана на бібліотеці `PyInstaller` та надає графічний інтерфейс, що робить процес створення EXE-файлу більш простим для користувача.

### 3.4.1 Основні функції які були реалізовані

Розроблена програма реалізує систему планування маршруту роботизованої платформи з використанням досконаленого алгоритму колонії мурах. Вона дозволяє генерувати карту з випадковими перешкодами або будувати лабіринт. Також є можливість робити та завантажувати власні мапи але вони повинні бути зроблені у CSV файлі та розміром 20x20. Результат роботи являється візуалізованим і ми можемо побачити весь маршрут роботизованої платформи. Для комфорту користувача був доданий користувацький інтерфейс.

Серед переваг мого програмного забезпечення можу назвати такі пункти:

- гнучкість у виборі мапи – існує два режими для побудови мапи: генерація мапи з випадковими перешкодами та створення лабіринту. Також є підтримка завантаження власних карт перешкод. Це дозволяє тестувати алгоритм на реальних картах;

- евристика – початкові маршрути створюються за допомогою евристичних алгоритмів, що підвищує якість маршруту;

Основні функції використані у роботі:

- `generate_valid_map()` – генерує випадкову карту з перешкодами, буферними зонами та перевіряє прохідність;
- `generate_maze_with_multiple_paths()` – генерує лабіринт з кількома можливими шляхами;
- `is_free(cell, obstacle_map, allow_buffer=False)` – перевіряє чи комірка вільна від різних перешкод;
- `allow_buffer=True` – дозволяє проходити через зони буферу навколо перешкод;
- `clear_zone_around(cell, obstacle_map, radius=2)` – очищає комірки у радіусі навколо точки за для того, щоб старт і фініш завжди були доступні;
- `bfs_check_path(start, goal, obstacle_map)` – перевіряє, чи існує шлях від старту до фінішу;
- `smart_path()` – розумний пошук шляху на карті з перешкодами;
- `smart_path_maze()` аналогічно до `smart_path()`, але працює для лабіринтів;
- `path_has_collision(path)` – перевірка того, чи проходить шлях через різні перешкоди або буферні зони;
- `fitness(path)` – фітнес функція для алгоритму мурашиної колонії за для покращення маршруту;
- `load_map_from_file()` – вибір файлу CSV, читає мапу з файлу та перевіряє її розмір;
- `draw_result()` – використовує `matplotlib` за для малювання карти, найкращого знайденого шляху, старту та фінішу;
- `start_gui(i)` – користувацький інтерфейс з кнопками генерування карти, завантаження карти та чекбоксу лабіринту (Maze);

Приклади роботи програми наведено на рис. 3.3 – рис 3.6.

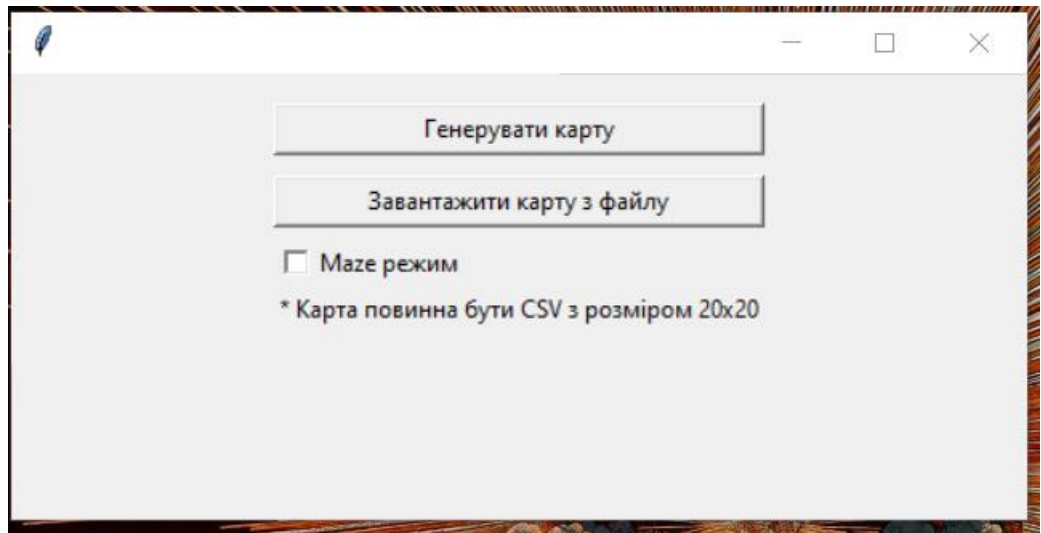


Рисунок 3.3 – Корстувачий інтерфейс розробленого програмного забезпечення

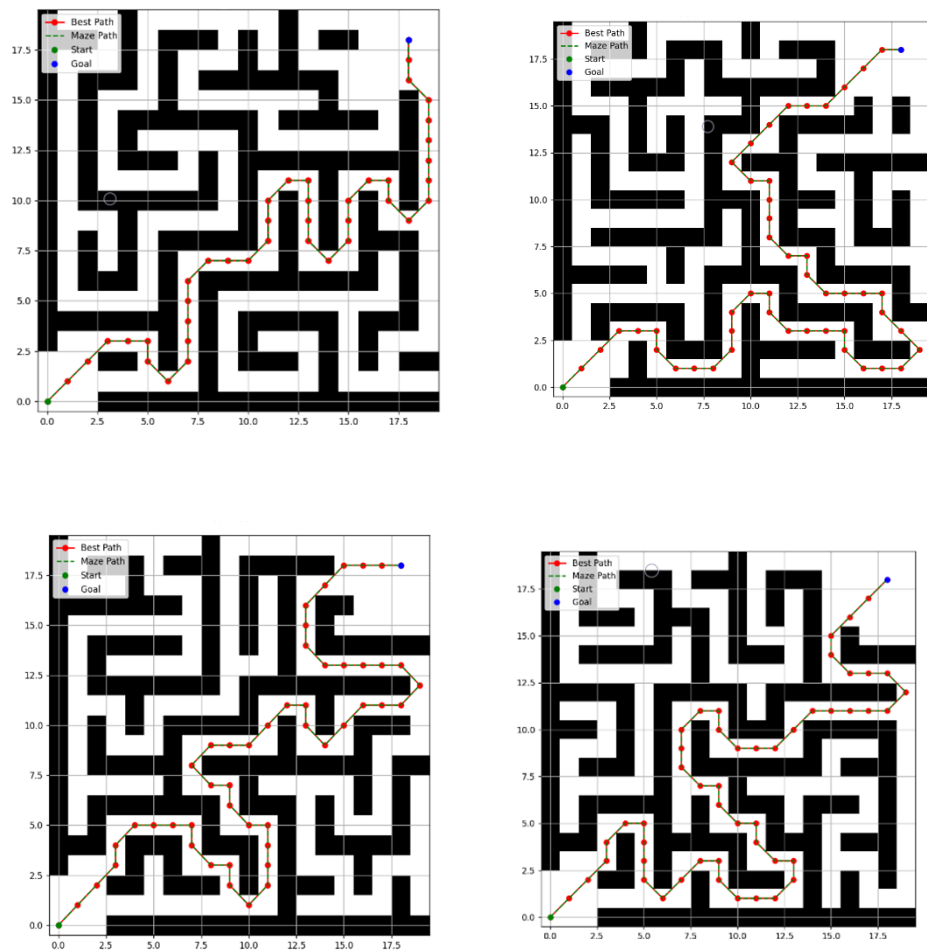


Рисунок 3.4 – Генерація лабіринту та проходження крізь нього

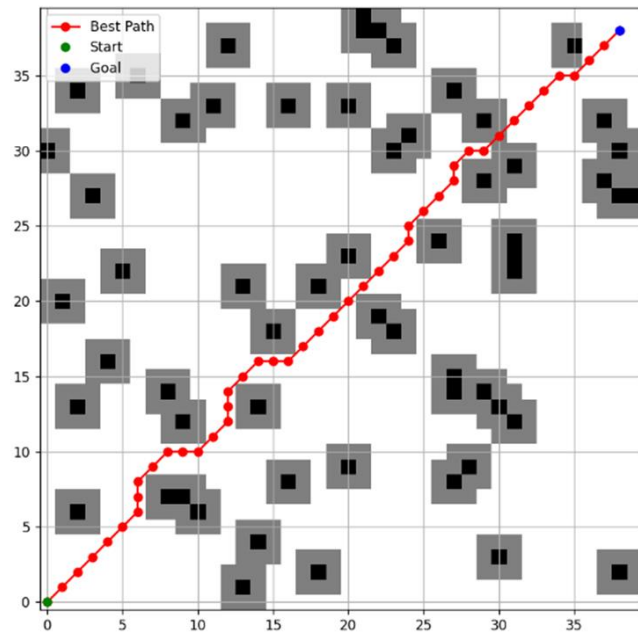


Рисунок 3.5 – Випадкова генерація перешкод та побудова маршруту

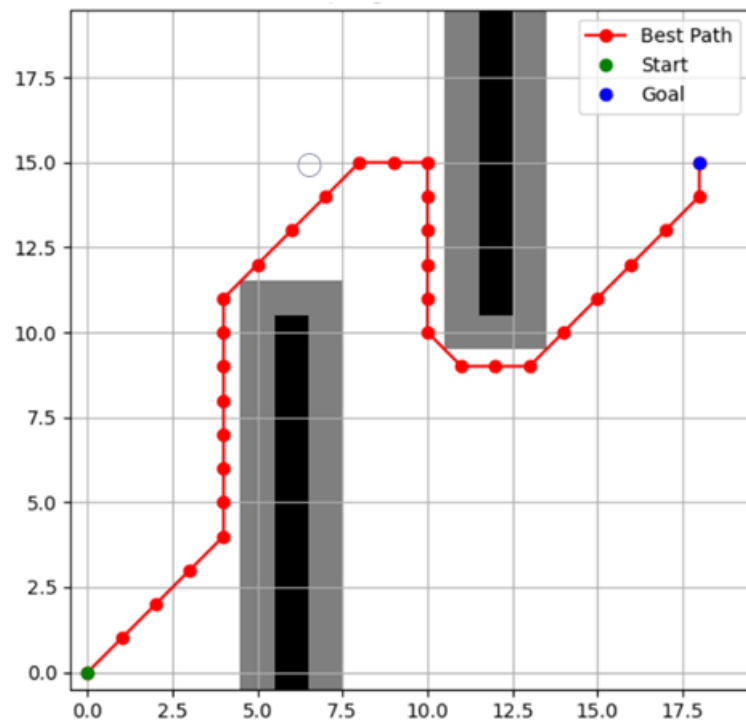


Рисунок 3.6 – Проходження карти створеної користувачем програмного забезпечення

### 3.5 Забезпечення безпечних умов праці при розробці автоматизованої системи

У ході виконання проєкту значна частина часу була присвячена роботі за комп'ютером, а саме у середовищу програмування. Цей вид діяльності пов'язаний із тривалим навантаженням і саме тому важливо було створити комфортні та безпечні умови праці [24].

Одним із головних факторів ефективної роботи є належне освітлення робочої зони. Щоб зменшити напруження очей, робота була виконана при м'якому розсіяному світлі, яке не створювало відблисків на екрані та не викликало різких перепадів яскравості в полі зору. Також враховувалося використання настільної лампи з теплою світловою температурою, що сприяло зниженню втоми під час тривалих сеансів роботи за персональним комп'ютером [24].

Не менш важливим аспектом було ергономічне облаштування робочого місця. Монітор розташовувався на рівні очей, що дозволяло уникати нахилів голови та зменшити навантаження на шийний відділ хребта. Регулярні перерви з розминкою забезпечували відпочинок очам і активність м'язам, що дозволяло запобігати типовим проблемам при тривалому сидінні в одному положенні – зокрема, м'язовій втомі, болю у спині та зниженню концентрації.

Оскільки програмування потребує високої концентрації, значну увагу приділено налаштуванню обладнання — зокрема, яскравості та контрастності дисплея, а також стабільності роботи програмного забезпечення. Це дозволяло забезпечити безперервну та зосереджену роботу без технічних збоїв.

Такі чинники, як належне освітлення, правильна постава, регулярні перерви та ергономіка, є невід'ємною частиною безпечної й результативної діяльності при створенні технічних пристроїв. Завдяки дотриманню зазначених умов вдалося не лише ефективно організувати робочий процес, а й підтримувати гарне самопочуття протягом усього періоду виконання проєкту.

Під час роботи особливу увагу було приділено дотриманню основних

рекомендацій щодо тривалого використання комп'ютера, оскільки процес створення програмного забезпечення вимагав високої концентрації протягом тривалого часу. Важливою умовою стало використання регульованого крісла з опорою для попереку та відповідним нахилом сидіння, що забезпечувало зручне положення тіла – спина мала опору, плечі були розслаблені, а зап'ястя не перенапружувалися [24].

Крім того, яскравість екрана налаштовувалась відповідно до освітленості кімнати. Також використовувався режим «нічного освітлення» для зменшення впливу синього світла, яке негативно діє на очі при роботі в темний час доби.

Окрему увагу приділяли чистоті робочої зони та обладнання. Регулярне очищення екрана, клавіатури та миші стало невід'ємною частиною організації робочого процесу.

Загалом, дотримання базових принципів гігієни праці за комп'ютером сприяло не лише збереженню здоров'я, а й підвищенню ефективності під час створення програмного забезпечення.

## ВИСНОВКИ

В результаті виконання кваліфікаційної роботи було досягнуто поставленої мети, а саме розроблено програмне забезпечення для автоматизованого визначення мінімальної траєкторії руху роботизованої платформи в умовах статичного та динамічного середовища. Реалізована система дозволяє враховувати фактори невизначеності, забезпечуючи надійність, точність та ефективність побудови маршрутів. У процесі дослідження було проведено аналіз сучасних методів планування шляхів для мобільних роботів, зокрема класичних (дорожні карти, розкладання комірки, штучні потенційні поля) та евристичних алгоритмів. Обґрунтовано вибір гібридного підходу, що поєднує алгоритм колонії мурах із методом динамічного вікна, що дозволяє уникати локальних мінімумів, зменшити кількість зайвих поворотів та підвищити швидкість збіжності. У роботі створено систему картографування та планування на основі мультисенсорного злиття даних (SLAM), що включає використання LiDAR, камери глибини та інерційних датчиків. Завдяки інтеграції цих компонентів, система здатна формувати точні карти оточення у реальному часі та планувати траєкторії, оминаючи перешкоди з урахуванням змін у середовищі. Практична реалізація програмного модуля показала працездатність системи та її здатність будувати оптимальні маршрути у заданих умовах. Особлива увага була приділена ергономіці та безпеці праці при розробці, що дозволило зберегти високу продуктивність та якість виконання проекту. Отримані результати можуть застосовуватися практично в логістичних, виробничих чи складських середовищах, де потрібна автономна навігація роботизованих платформ.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Ільєнков Г. О. Аналіз алгоритмів планування шляху мобільного робота / Г. О. Ільєнков // Автоматизація та приладобудування («Automation and Development of Electronic Devices» ADED-2025) : збірник студентських наукових статей. Харків : ХНУРЕ, 2025. Вип. 1. С. 83–87.

2. Навчальний посібник з підготовки кваліфікаційної роботи бакалавра для здобувачів вищої освіти денної і заочної форм навчання спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» освітньої програми «Автоматизація та комп'ютерно-інтегровані технології» : Навчальний посібник / І. Ш. Невлюдов, О. І. Филипченко, О. В. Токарева, С. П. Новоселов, О. В. Сичова. Харків : Видавництво Іванченка І. С., 2023. – 151 с.

3. Методичні вказівки з підготовки кваліфікаційної роботи для здобувачів першого (бакалаврського) рівня вищої освіти денної і заочної форми навчання спеціальності 151 – Автоматизація та комп'ютерно-інтегровані технології освітньої [Електронний ресурс] / упоряд.: І. Ш. Невлюдов, О. І. Филипченко, О. В. Токарева, С. П. Новоселов, О. В. Сичова ; М-во освіти і науки України, Харків. нац. ун-т радіоелектроніки. електрон. вид. Харків : ХНУРЕ, 2023. 760 Кб

4. ДСТУ 3008-15. Документація. Звіти у сфері науки та техніки. структура та правила оформлення. Введ. 2015-06-22. К. Держстандарт України, 2017. 29 с.

5. Основи розробки кросплатформного програмного забезпечення на Avalonia: Навчальний посібник / С. П. Новоселов, О. В. Сичова. Харків: Видавництво Іванченка І. С., 2024. 267 с.

6. Невлюдов І. Ш., Андрусевич А. О., Євсєєв В. В., Новоселов С. П., Демська Н. П. Проектування мобільних маніпуляційних роботів: Монографія. – Х. :, 2022. – 427 с. 7. Cognominal, M., Patronymic, K., & Wańkiewicz, A. (2021). Evolving field of autonomous mobile robotics: Technological advances and

applications. *Fusion of Multidisciplinary Research, An International Journal*, 2(2), 189-200.

8. Abdulsaheb, J. A., & Kadhim, D. J. (2023). Classical and heuristic approaches for mobile robot path planning: A survey. *Robotics*, 12(4), 93.

9. Sokkalingam, S., & Ramakrishnan, R. (2022). An intelligent intrusion detection system for distributed denial of service attacks: A support vector machine with hybrid optimization algorithm based approach. *Concurrency and Computation: Practice and Experience*, 34(27), e7334.

10. Kumar, N., Sinwar, D., Saini, M., Saini, D. K., Kumar, A., Kaur, M., ... & Lee, H. N. (2022). Efficient computational stochastic framework for performance optimization of E-waste management plant. *Journal of King Saud University-Computer and Information Sciences*, 34(8), 4712-4728.

11. Abdulsaheb, J. A., & Kadhim, D. J. (2022). Robot Path Planning in Unknown Environments with Multi-Objectives Using an Improved COOT Optimization Algorithm. *International Journal of Intelligent Engineering & Systems*, 15(5).

12. Abdulsaheb, J. A., & Kadhim, D. J. (2023). Multi-Objective Robot Path Planning Using an Improved Hunter Prey Optimization Algorithm. *International Journal of Intelligent Engineering & Systems*, 16(2).

13. Gabbassova, Z., Sedighizadeh, D., Sheikhi Fini, A., & Seddighizadeh, M. (2021). Multiple robot motion planning considering shortest and safest trajectory. *Electromechanical Energy Conversion Systems*, 1(3), 1-6.

14. Badmos, T. A., Omolaye, P. O., Mebawondu, J., & Aliyu, H. A. (2020). Robot path planning performance evaluation of a dynamic environment. *IOSR J. Electron. Commun. Eng.*, 13(6), 19-26.

15. Abdulsaheb, J. A., & Kadhim, D. J. (2022). Robot Path Planning in Unknown Environments with Multi-Objectives Using an Improved COOT Optimization Algorithm. *International Journal of Intelligent Engineering & Systems*, 15(5).

16. Abdulsaheb, J. A., & Kadhim, D. J. (2023). Multi-Objective Robot Path

Planning Using an Improved Hunter Prey Optimization Algorithm. *International Journal of Intelligent Engineering & Systems*, 16(2).

17. Thammachantuek, I., & Ketcham, M. (2022). Path planning for autonomous mobile robots using multi-objective evolutionary particle swarm optimization. *Plos one*, 17(8), e0271924.

18. Shi, Y., Zhang, H., Li, Z., Hao, K., Liu, Y., & Zhao, L. (2023). Path planning for mobile robots in complex environments based on improved ant colony algorithm. *Mathematical Biosciences and Engineering*, 20(9), 15568-15603.

19. Gao, W., Tang, Q., Ye, B., Yang, Y., & Yao, J. (2020). An enhanced heuristic ant colony optimization for mobile robot path planning. *Soft Computing*, 24, 6139-6150.

14. Wang, L. (2020). Retracted: Path planning for unmanned wheeled robot based on improved ant colony optimization. *Measurement and Control*, 53(5-6), 1014-1021.

20. Al Baroomi, B., Myo, T., Ahmed, M. R., Al Shibli, A., Marhaban, M. H., & Kaiser, M. S. (2023, August). Ant Colony Optimization-Based Path Planning for UAV Navigation in Dynamic Environments. In *2023 7th International Conference on Automation, Control and Robots (ICACR)* (pp. 168-173). IEEE.

21. Si, J., & Bao, X. (2024). A novel parallel ant colony optimization algorithm for mobile robot path planning. *Mathematical Biosciences and Engineering: MBE*, 21(2), 2568-2586.

22. Höffmann, M., Patel, S., & Büskens, C. (2024). Optimal guidance track generation for precision agriculture: A review of coverage path planning techniques. *Journal of Field Robotics*, 41(3), 823-844.

23. Cichella, V., Kaminer, I., Walton, C., Hovakimyan, N., & Pascoal, A. M. (2020). Optimal multivehicle motion planning using bernstein approximants. *IEEE transactions on automatic control*, 66(4), 1453-1467.

24. Комплекс навчально-методичного забезпечення навчальної дисципліни «Безпека праці в індустрії ІТ-технологій» підготовки освітнього рівня бакалавр усіх спеціальностей та усіх напрямів університету

[<http://catalogue.nure.ua/knmz>] / ХНУРЕ; розроб.: Т. Є. Стищенко, Г. В. Пронюк, Н. М. Сердюк. – Харків, 2017. – 122 с.