Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Факультет комп’ютерної інженерії та управління
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Алгоритм Lazy Theta* руху "s-bot"
з урахуванням рельєфу місцевості

(тема)

Виконав:

студент II курсу, групи СПм-22-3

Гриценко І.К.
(прізвище, ініціали)

Спеціальність
123 «Комп’ютерна інженерія»
(код і повна назва спеціальності)

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

Освітня програма
Системне програмування
(повна назва освітньої програми)

Керівник: доц. Токарєв В.В.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри ЕОМ _____ Коваленко А.А.
(підпис) (прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки

Факультет комп’ютерної інженерії та управління

Кафедра електронних обчислювальних машин

Рівень вищої освіти другий (магістерський)

Спеціальність 123 «Комп’ютерна інженерія»
(код і повна назва)

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

Освітня програма Системне програмування
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
“____”_____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студенту Гриценку Івану Костянтиновичу
(прізвище, ім’я, по батькові)

1. Тема роботи Алгоритм Lazy Theta* руху "s-bot" з урахуванням рельєфу місцевості

затверджена наказом по університету від “ 01 ” квітня 2024 р. № 257Ст

2. Термін подання студентом роботи до екзаменаційної комісії 15 червня 2024р.

3. Вхідні дані до роботи 1) провести огляд та аналіз методів навігації та позиціонування "s-bot" у physical unorganized environment; 2) провести огляд методів планування маршруту руха інтелектуального мобільного "s-bot" у physical unorganized environment; 3) провести дослідження метода SLAM; 4) провести дослідження алгоритму Lazy Theta*.

4. Перелік питань, що потрібно опрацювати у роботі
1) огляд літератури за темою роботи;
2) аналіз предметної області;
3) вибір та обґрунтування методики дослідження;
4) проведення експериментальних досліджень;
5) висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів)

Слайд-презентація – 20 слайдів

6. Консультанти розділів роботи (заповнюється 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	Огляд літератури за темою роботи	02.04.24 - 08.04. 24	
2	Вибір та обгрунтування методики дослідження	09.04.24 - 16.04.24	
3	Вибір інструментальних засобів	17.04.24 - 22.04.24	
4	Розробка моделей протоколів	23.04.24 - 06.05.24	
5	Проведення експериментів	07.05.24 - 23.05.24	
6	Оформлення матеріалів кваліфікаційної роботи	24.05.24-03.06.24	
7	Подання кваліфікаційної роботи керівникові та	04.06.24-	
	її попередній захист	07.06.24	
8	Подання кваліфікаційної роботи на	08.06.24-	
	рецензування	12.06.24	

Дата видачі завдання 01 квітня 2024 р.

Студент _____
(підпис)

Керівник роботи _____ доц. Токарєв В.В.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 69 с., 36 рис., 1 табл., 1 дод., 14 джерел.

АЛГОРИТМ LAZY THETA*, МАТЕМАТИЧНА МОДЕЛЬ, УПРАВЛІННЯ РУХОМ, ФІЗИЧНЕ НЕОРГАНІЗОВАНЕ СЕРЕДОВИЩЕ, "S-BOTS"

Метою кваліфікаційної роботи є дослідження алгоритму Lazy Theta*, для його застосування в системах навігації та управління рухом інтелектуальним мобільним "s-bot", що входить до складу однієї "Swarm - bot" - system у physical unorganized environment.

У ході виконання кваліфікаційної роботи досліджується можливість планування маршрутів руху інтелектуального мобільного "s-bot", що входить до складу однієї "Swarm - bot" - system у physical unorganized environment. Вирішення поставленого завдання передбачає дослідження алгоритму Lazy Theta*. В даний час одна з найважливіших проблем, що вимагає вирішення при плануванні маршруту руху інтелектуального мобільного "s-bot", полягає в тому, що його траєкторія руху від початкової до кінцевої (цільової) точки маршруту в загальному випадку проходить в маловивченій обстановці. Але ця траєкторія руху повинна забезпечувати гарантовану безпеку "s-bot". Тому дослідження алгоритму планування маршруту інтелектуального мобільного "s-bot", що входить до складу однієї "Swarm - bot" - system, який у режимі реального часу забезпечував би пошук короткої та безпечної траєкторії, є в даний час актуальним завданням.

ABSTRACT

Master’s thesis: 69 pages, 36 figures, 1 tables ,1 appendices, 14 sources.

LAZY THETA* ALGORITHM, MATHEMATICAL MODEL, MOTION CONTROL, PHYSICAL UNORGANIZED ENVIRONMENT, "S-BOTS"

The purpose of the qualification work is to study the Lazy Theta* algorithm for its application in the navigation and motion control systems of an intelligent mobile "s-bot", which is part of one "Swarm - bot" - system in a physical unorganized environment.

In the course of the qualification work, the possibility of planning the movement routes of an intelligent mobile "s-bot", which is part of one "Swarm - bot" - system in a physical unorganized environment, is investigated. Solving the task involves researching the Lazy Theta* algorithm. Currently, one of the most important problems that requires a solution when planning the route of movement of an intelligent mobile "s-bot" is that its trajectory from the initial to the final (target) point of the route generally takes place in a poorly studied environment. But this trajectory of movement should ensure the guaranteed safety of "s-bot". Therefore, the research of the route planning algorithm of the intelligent mobile "s-bot", which is part of one "Swarm - bot" - system, which would ensure the search of a short and safe trajectory in real time, is currently an urgent task.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ 8
ВСТУП 9
1 ОГЛЯД СУЧАСНОГО СТАНУ РІШЕННЯ ЗАДАЧ НАВІГАЦІЇ І ПЛАНУВАННЯ МАРШРУТУ РУХУ ІНТЕЛЕКТУАЛЬНОГО МОБІЛЬНОГО "S-BOT" 11
1.1 Огляд методів навігації та позиціонування інтелектуального мобільного "s-bot" у physical unorganized environment 12
1.2 Аналіз методу SLAM 16
1.2.1 Аналіз рішення задачі SLAM за допомогою лазерного далекоміра 22
1.2.2 Аналіз вирішення задачі SLAM за допомогою відеокамер VSLAM 25
1.3 Задача планування маршруту руху інтелектуальних мобільних "s-bots", що входять до складу однієї "Swarm-bot"-system у physical unorganized environment 28
2 АЛГОРИТМ LAZY THETA* ПЛАНУВАННЯ МАРШРУТУ РУХУ ІНТЕЛЕКТУАЛЬНОГО МОБІЛЬНИЙ "S-BOT" З УРАХУВАННЯМ РЕЛЬЄФУ МІСЦЕВОСТІ 31
2.1 Огляд алгоритмів пошуку маршруту руху інтелектуального мобільного "s-bot" у physical unorganized environment 32
2.1.1 Огляд алгоритму A* 32
2.1.2 Огляд Алгоритму планування руху в будь-якому напрямку 37
2.1.3 Огляд алгоритму Basic Theta* 39
2.2 Дослідження алгоритму Lazy Theta* 44
3 ДОСЛІДЖЕННЯ АЛГОРИТМУ LAZY THETA* РУХУ "S-BOT" З УРАХУВАННЯМ РЕЛЬЄФУ МІСЦЕВОСТІ 50
ВИСНОВКИ 55

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ 56
ДОДАТОК А Графічний матеріал кваліфікаційної роботи..... 59

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ВІС – візуально-інерційна система

ШНМ – штучна нейронна мережа

EKF – розширений фільтр Калмана (англ., Extended Kalman Filter)

SLAM – метод одночасної локалізації та побудови карти (англ., Simultaneous Localization And Mapping)

VSLAM – візуальний SLAM (англ., Visual Simultaneous Localization And Mapping)

UKF – сигма-точковий фільтр Калмана (англ., Unscented Kalman filter);

ВСТУП

При проведенні наукових досліджень у physical unorganized environment із застосуванням платформи "Swarm-bot"-system, до складу якої входять інтелектуальні мобільні "s-bot", виникають проблеми пов'язані з:

- складністю рельєфу невідомої чи маловивченої місцевості;
- обмеженнями на наявні запаси бортових джерел енергії на "s-bot";
- затримками чи відсутністю комунікації "s-bot" з оператором наземного пункту управління;
- можливістю зміни основних об'єктів наукових досліджень у ході місії.

При використанні лише дистанційного керування інтелектуальним мобільним "s-bot", неможливо досягти ефективної навігації, тому не всі вищезгадані обмеження можна подолати.

Крім того, дистанційне управління при дослідженні physical unorganized environment відмінної від земної поверхні, стає непрактичним через тривалий період передачі інформації та прийняття рішень.

Щоб забезпечити безпечні умови пересування інтелектуального мобільного "s-bot" в physical unorganized environment, необхідно в реальному режимі часу визначати положення інтелектуального мобільного "s-bot", щодо перешкод та заданих цільових точок, своєчасно отримувати інформацію про місцевість, яка оточує інтелектуальний мобільний "s-bot" та вирішувати завдання позиціонування. Поєднання завдань позиціонування та навігації інтелектуального мобільного "s-bot" із завданням побудови карти місцевості необхідно для обчислення наближення до заданої мети та ухилення від перешкод, які можуть виникнути на маршруті руху. В даний час до системи навігації та управління рухом інтелектуальним мобільним "s-bot" пред'являються вимоги пов'язані з точністю та позиціонуванням інтелектуального мобільного "s-bot" у реальному режимі часу. При вирішенні

завдання позиціонування тільки за даними, що надходять з відеокамери, виникають складності, пов'язані з обертальним рухом рухомого об'єкта, його підвищеною лінійною швидкістю руху, можливою відсутністю характерних точок сцен.

Тому розробка алгоритму в системах навігації та управління рухом інтелектуальних мобільних "s-bots", що входять до складу однієї "Swarm-bot" -system у physical unorganized environment є актуальною науковою задачею.

1 ОГЛЯД СУЧАСНОГО СТАНУ РІШЕННЯ ЗАДАЧ НАВІГАЦІЇ І ПЛАНУВАННЯ МАРШРУТУ РУХУ ІНТЕЛЕКТУАЛЬНОГО МОБІЛЬНОГО "S-BOT"

У загальному вигляді завдання керування рухом інтелектуальним мобільним "s-bot" можна сформулювати в такий спосіб:

- крок № 1 – визначення свого становища в physical unorganized environment в режимі реального часу;

- крок № 2 – побудова чи оновлення карти physical unorganized environment в режимі реального часу;

- крок № 3 – планування маршруту руху інтелектуального мобільного "s-bot";

- крок № 4 – необхідність обходу перешкод інтелектуальним мобільним "s-bot" та запобігання зіткненню з ними.

На перший погляд, кроки здаються простими, проте існує ціла низка проблем, умов та обмежень, серед яких основними є такі кроки:

- крок № 1 – втрата комунікації з датчиками, які здійснюють взаємодію з physical unorganized environment;

- крок № 2 – обмежені обчислювальні та енергетичні ресурси інтелектуального мобільного "s-bot";

- крок № 3 – втрата комунікації з "Swarm-bot"-system.

Таким чином, основна проблема автономного інтелектуального мобільного "s-bot" полягає в складності автоматичного обходу різних перешкод, автономної навігації, втрати комунікації з датчиками, які здійснюють взаємодію з physical unorganized environment та втрати комунікації з базовою платформою "Swarm-bot"-system. В даний час в економічно розвинених країнах ведуться дослідження з розробки Nanorovers (рисунок 1.1), які виступатимуть у ролі інтелектуальних мобільних "s-bots", що входять до складу однієї "Swarm-bot"-system.

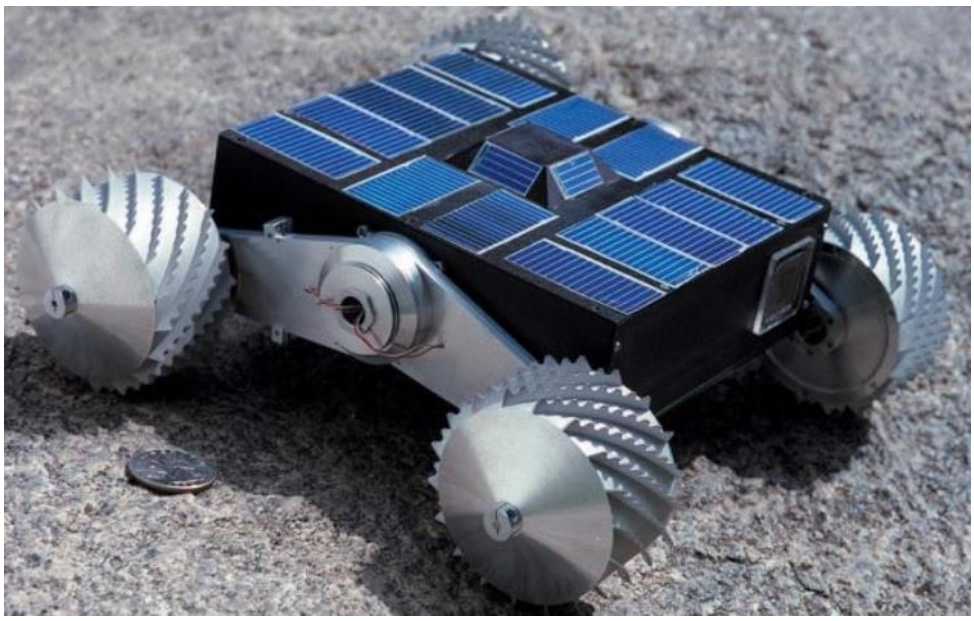


Рисунок 1.1 – Приклад Nanorover

Nanorovers розробляються на допомогу астронавтам для виконання поставлених перед експідицією завдань. Це сприяє розширенню діапазону досліджень та безпеки життя астронавтів, тоді як самі астронавти залишаються у безпечній зоні. У найближчому майбутньому Nanorovers з вантажопідйомністю від 10 кг до 50 кг будуть застосовуватися для локального дослідження в діапазоні до 1 км. Тому застосування Nanorovers дозволить підвищити ефективність та безпеку виконання поставлених завдань.

1.1 Огляд методів навігації та позиціонування інтелектуального мобільного "s-bot" у physical unorganized environment

Система позиціонування та навігації інтелектуального мобільного "s-bot" є ключовою частиною системи управління його руху. Позиціонування та навігація у сучасних інтелектуальних мобільних "s-bots" реалізовані за допомогою різних методів, таких як:

- метод зчислення шляху або координат;

- метод візуальної одометрії;
- метод автокалібрування камери з алгоритмом налаштування Bundle adjustment.

Метод зчислення шляху або координат на основі інерційної навігаційної системи обчислює положення інтелектуального мобільного "s-bots". Перевага методу зчислення шляху або координат полягає в тому, що він має простий розрахунок та невелику споживану потужність обчислення. При використанні методів зчислення шляху для вирішення навігаційних задач використовуються датчики, що вимірюють похідні від параметрів навігації. Нехай є деякий початковий момент часу, в який навігаційні параметри, зокрема координати, відомі. Інтегруючи вимірювання похідних координат (швидкості, прискорення і т.д.) з початкової точки, одержують вимірювання самих координат (рисунок 1.2).

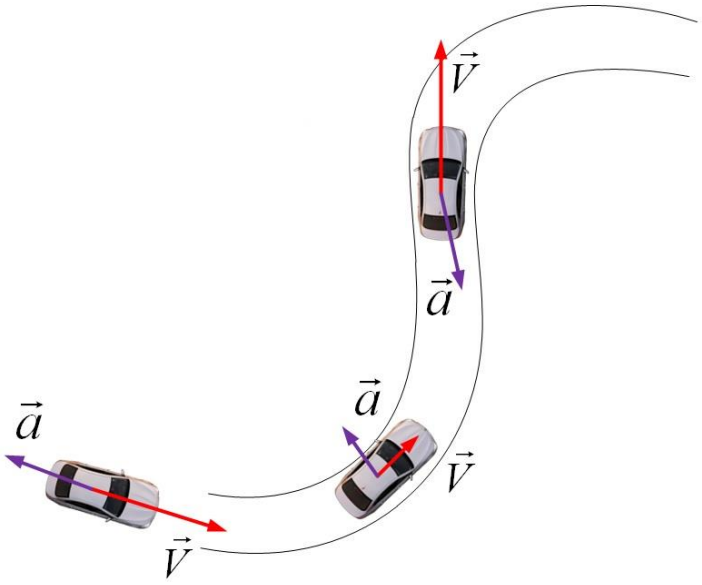


Рисунок 1.2 – Приклад використання методу зчислення шляху для вирішення навігаційних задач

Недоліком методу зчислення шляху або координат є те, що в процесі переміщення інтелектуального мобільного "s-bots" за рахунок пробуксування коліс відбувається накопичення помилки.

Метод візуальної одометрії – visual odometry полягає в тому, щоб виправити помилку позиціонування, викликану методом зчислення шляху або координат при пробуксуванні коліс (рисунок 1.3).

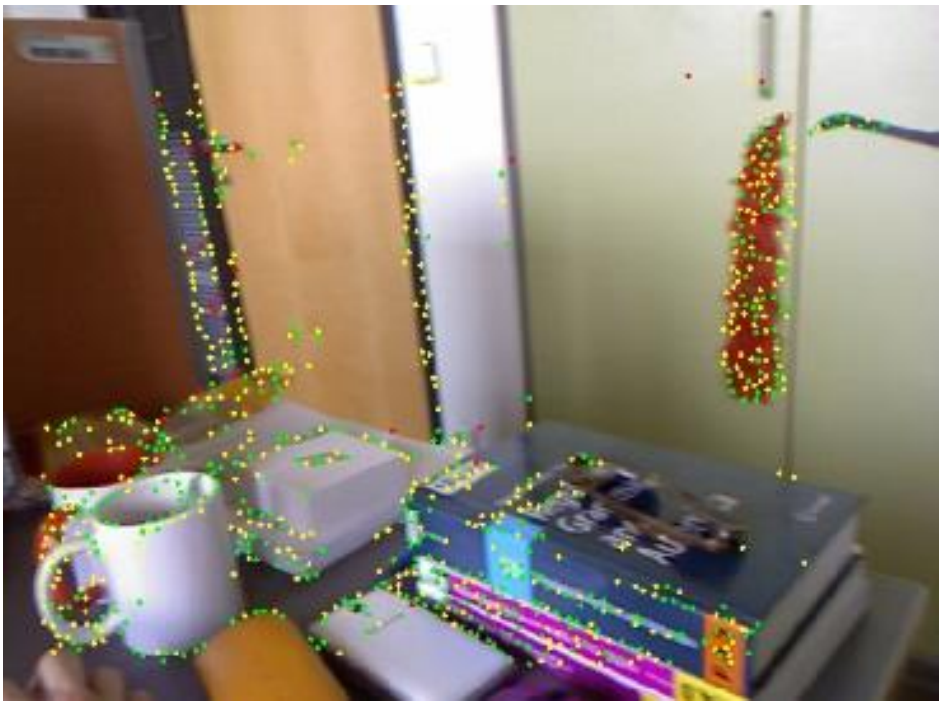


Рисунок 1.3 – Приклад використання методу візуальної одометрії для вирішення навігаційних задач

Стандартний алгоритм, розроблений на основі методу візуальної одометрії, складається з наступних кроків:

- крок № 1 – отримання зображення з камери;
- крок № 2 – корекція зображення (усунення дісторсії);
- крок № 3 – детектування ключових точок зображення методом Lucas- Kanade;
- крок № 4 – порівняння точок між кадрами;
- крок № 5 – побудова оптичного потоку;
- крок № 6 – перевірка векторів оптичного потоку на потенційні помилки;

- крок № 7 – визначення руху камери з оптичного потоку, використовуючи фільтр Калмана;

- крок № 8 – періодичне оновлення набору ключових точок для відстеження (рисунок 1.3).

Недоліком методу візуальної одометрії є те, що через обмеження швидкості комунікації між інтелектуальним мобільним "s-bots" з базовою платформою "Swarm-bot"-system, потрібно близько 3 хвилин, щоб отримати, обробити дані та оновити положення. Тому метод візуальної одометрії не може застосовуватися під час руху інтелектуального мобільного "s-bot" самостійно.

Цей метод застосовується лише при обчисленні локального позиціонування, коли інтелектуальний мобільний "s-bot" наближається до зазначеної мети і йому залишається не більше 15 м.

Метод автокалібрування камери з алгоритмом налаштування Bundle adjustment полягає в тому, що він не вимагає безперервної фотозйомки на коротких відстанях (рисунок 1.4).

Рисунок 1.4 – Приклад використання методу автокалібрування камери для вирішення навігаційних завдань

Точність позиціонування цього методу є високою. Недоліком цього методу є те, що він потребує великої кількості обчислювальної потужності бортової апаратури інтелектуального мобільного "s-bot".

Слід зазначити, що при дистанційному управлінні інтелектуальним мобільним "s-bot", потрібна участь людини-оператора. Головна проблема полягає в тому, що за допомогою використання лише одного із зазначених методів не можна забезпечити задані вимоги щодо точності автономної навігації. З появою у 2006 році методу одночасної локалізації та побудови карти – simultaneous localization and mapping (SLAM) – та його подальшого розвитку стало можливо реалізувати повністю автономний режим позиціонування та навігації інтелектуальних мобільних "s-bots", що входять до складу однієї "Swarm-bot"-system.

1.2 Аналіз методу SLAM

Метод SLAM – метод, який використовується в інтелектуальних мобільних "s-bots", що входять до складу однієї "Swarm-bot"-system для побудови карти в physical unorganized environment або для оновлення карти в заздалегідь відомому просторі з одночасним контролем поточного розташування та пройденого шляху.

На сьогоднішній день існують наступні різновиди алгоритмів, розроблених на методі SLAM.

Алгоритм № 1 – алгоритм Feature-based SLAM (рисунок 1.5). Цей алгоритм використовує легко ідентифіковані сутності в physical unorganized environment і створює внутрішнє уявлення про навколишній простір з урахуванням розташування цих сутностей. Сам алгоритм заснований на розширеному фільтрі Калмана – Extended Kalman Filter (EKF).

Алгоритм № 2 – алгоритм Graph-based SLAM (рисунок 1.6). Цей алгоритм заснований на теорії графів або мереж, також як попередній алгоритм, створює карту за допомогою графа, вузли якого відповідають

позиціям "s-bots" у різні моменти часу, а ребра представляють собою просторові обмеження, що зв'язують положення "s-bots" разом.

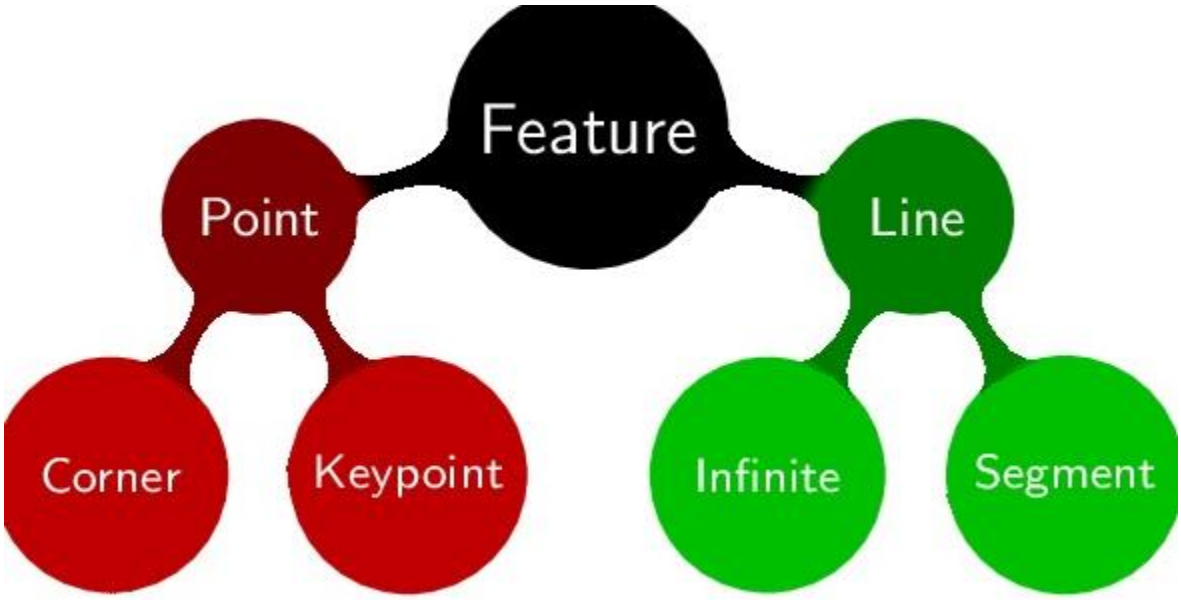


Рисунок 1.5 – Алгоритм Feature-based SLAM

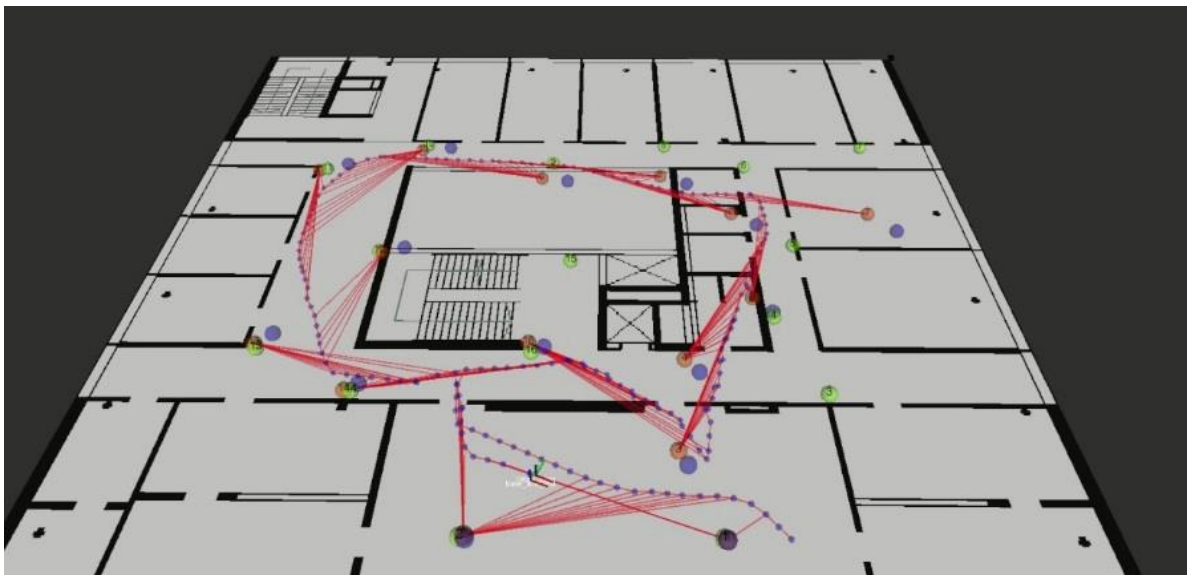


Рисунок 1.6 – Алгоритм Graph-based SLAM

Алгоритм № 3 – алгоритм Grid-based SLAM (рисунок 1.7). Цей алгоритм є теоретично найбільш простим підходом з урахуванням сітки.

Простір розбивається на сітку крапок із певним рівним кроком. Кожна точка може мати свій стан: зайнято, вільно, перешкода.

Рисунок 1.7 – Алгоритм Grid-based SLAM

Алгоритм № 4 – алгоритм Topological SLAM (рисунок 1.8). Цей алгоритм вирішує поставлені перед ним завдання за допомогою графоподібного опису навколишнього середовища, а не точної метричної карти. Під топологічним описом розуміються вузлові місця, які можна легко порівняти та відрізнити.

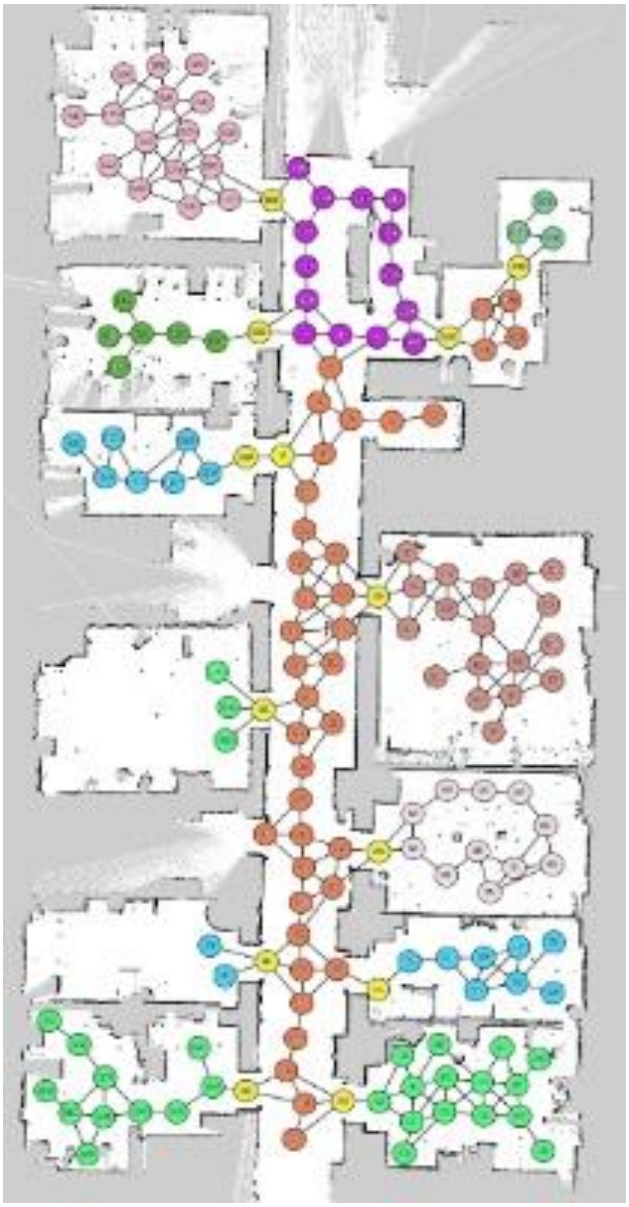


Рисунок 1.8 – Алгоритм Topological SLAM

Алгоритм № 5 – алгоритм Semantic SLAM (рисунок 1.9). У цьому алгоритмі карти представлені як моделі семантичних карт. На відміну від топологічних методів, у яких алгоритми фільтрують метричну інформацію та використовують лише розпізнавання місць-орієнтирів для визначення місця розташування, семантичне зіставлення пов'язує семантичні концепції з об'єктами в physical unorganized environment.

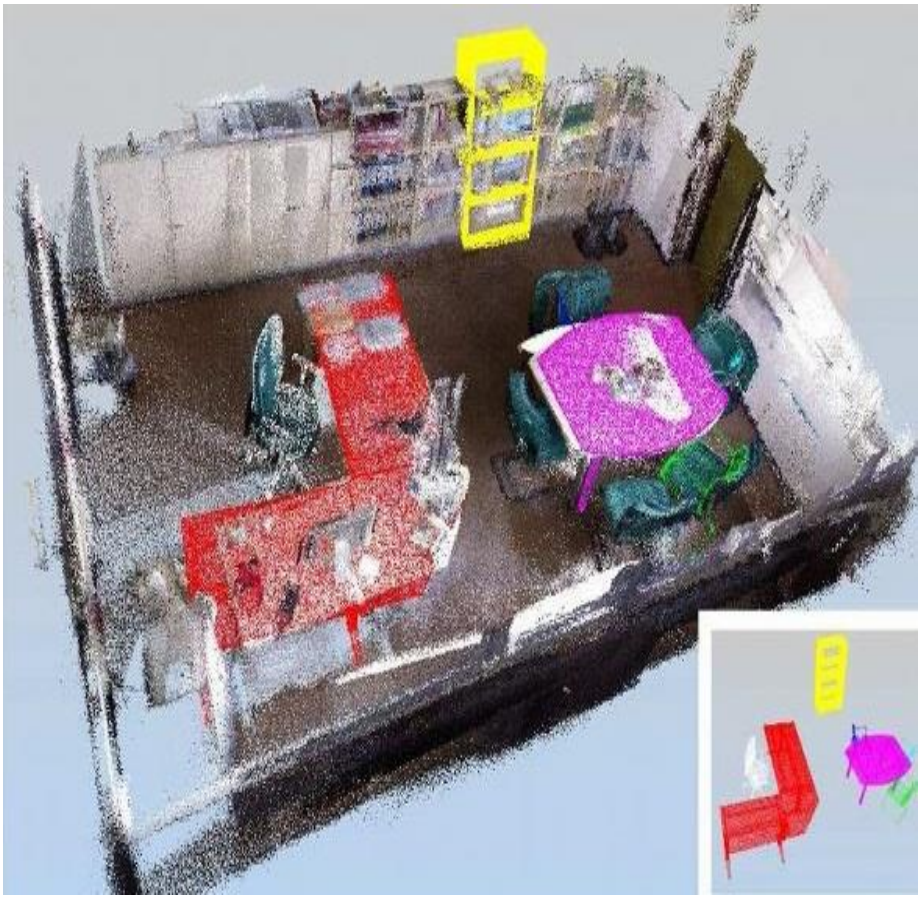


Рисунок 1.9 – Алгоритм Semantic SLAM

Метод SLAM широко застосовується в:

- безпілотних наземних засобах;
- безпілотних повітряних засобах;
- безпілотних підводних засобах;
- планетоходах.

Ця технологія стала основною у навігації інтелектуальних мобільних "s-bots", що входять до складу однієї "Swarm-bot"-system.

Загальна структура методу SLAM із плануванням маршруту представлена на рисунку 1.10. Загалом, система SLAM зазвичай включає безліч датчиків і кілька функціональних модулів.

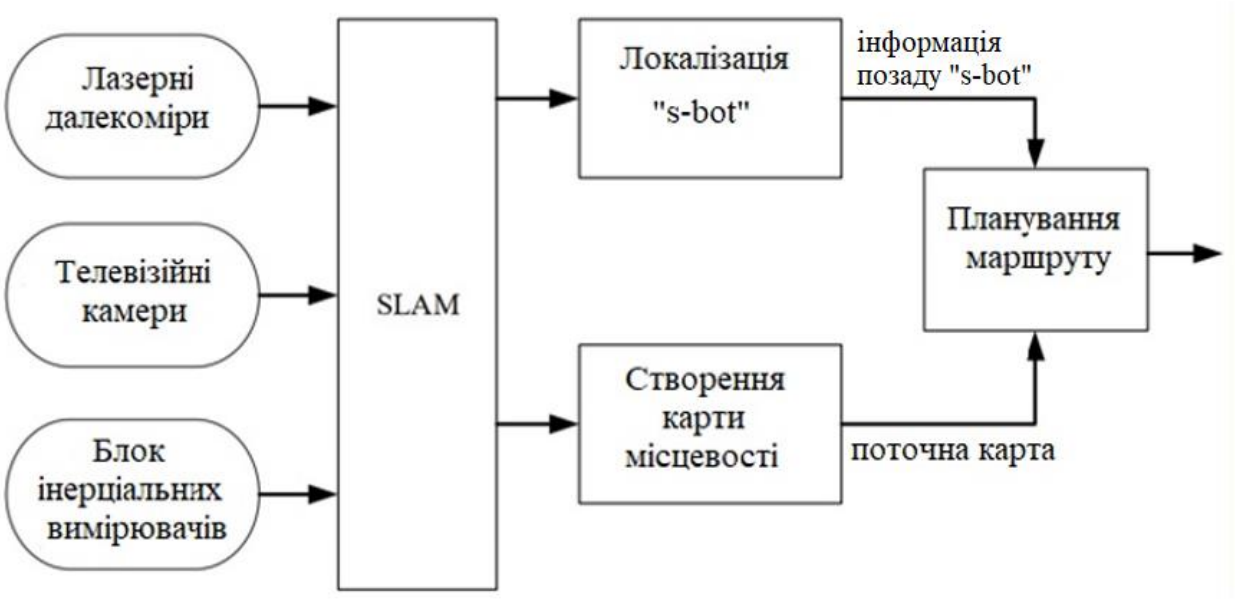


Рисунок 1.10 – Загальна структура методу SLAM із плануванням маршруту

Системний датчик (включає в себе лазерні далекоміри, телевізійні камери та інерціальні вимірювачі, тощо) отримує кодовані дані від сенсора і перетворює їх у зрозумілий для подальшої обробки вектор виміряних відстаней. Фільтр хибних точок видаляє хибні (некоректні) виміри. Також він перетворює точки, що залишилися, в декартову систему координат для подальшої обробки. Під час планування маршруту інтелектуальних мобільних "s-bots", що входять до складу однієї "Swarm-bot"-system, використовується поточне положення інтелектуального мобільного "s-bot" і карта, яка була отримана за методом SLAM, щоб прокласти оптимальний маршрут руху інтелектуального мобільного "s-bot" до цільової точки.

За типом, що використовуються в сучасних навігаційних системах датчиків, які встановлені на сучасних інтелектуальних мобільних "s-bots" розрізняють два варіанти розв'язання задачі SLAM:

- лазерний SLAM;
- візуальний SLAM - VSLAM.

Поява нових модифікацій методу SLAM доводить його живучість для розв'язання задач навігації інтелектуальних мобільних "s-bots". Застосування

методу SLAM для вирішення задач автономної навігації інтелектуальних мобільних "s-bots" поки що недостатньо добре вивчене. Тому дослідження методу SLAM для його застосування в системах навігації та управлінні рухом інтелектуальних мобільних "s-bots", що входять до складу однієї "Swarm-bot"-system у physical unorganized environment є актуальним завданням.

1.2.1 Аналіз рішення задачі SLAM за допомогою лазерного далекоміра

Розв'язання задачі SLAM, за допомогою використання лазерних далекомірів, є відносно стабільним способом, який має високу точність, невелику накопичувальну помилку і створює інтуїтивно зрозумілу карту. При такому підході метод SLAM можна розділити на дві складові:

- складова frontend-SLAM;
- складова backend-SLAM.

Складова fronted-SLAM включає добування особливостей feature extraction і асоціацію даних data association. Під добуванням параметрів, що часто зустрічаються у просторі, розуміються:

- кути стін;
- прямі лінії;
- контрастні точки (для відеокамери).

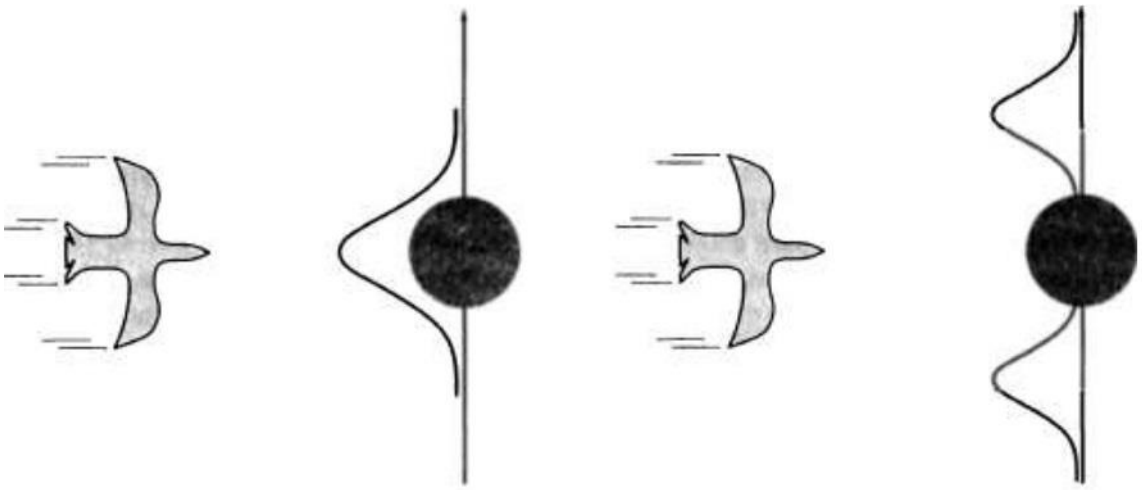
Асоціація даних – це процес для перетворення отриманих даних від сенсорів до уніфікованого уявлення, яке подається на вхід безпосередньо SLAM-ядру.

Складова backend-SLAM – це безпосередня реалізація розв'язання задачі SLAM, оптимізація отриманих від складової fronted-SLAM просторових відносин з метою максимізації ймовірності становища інтелектуальних мобільних "s-bots", що входять до складу однієї "Swarm-bot"-system в physical unorganized environment, за допомогою алгоритмів фільтрації. Складова backend-SLAM є предметом досліджень у

кваліфікаційній роботі. В даний час для вирішення задачі backend-SLAM за допомогою лазерного далекоміра існує великий вибір алгоритмів, що використовують наближені методи обчислень, які включають спеціальну фільтрацію:

- фільтр Калмана;
- розширений фільтр Калмана (EKF);
- сигма-точковий фільтр Калмана (UKF);
- фільтр частинок (PF).

Використовують спеціальну фільтрацію EKF та UKF для того, щоб отримати апостеріорний розподіл положень інтелектуальних мобільних "s-bots", що входять до складу однієї "Swarm-bot"-system у physical unorganized environment, щодо орієнтирів або перешкод. Приклад приведений на рисунку 1.11.



а) б)

Рисунок 1.11 – Приклад із птахом, що летить прямо на дерево (вид зверху): а) фільтр Калмана передбачає місцезнаходження птаха з використанням єдиного гауссового розподілу, центр якого знаходиться навпроти перешкоди; б) більш реалістична модель допускає виконання птахом дії щодо ухилення від зіткнення з перешкодою, передбачаючи, що птах облетить перешкоду з одного боку чи з іншого

Однак, спеціальна фільтрація EKF передбачає лінеаризацію матриці

переходу та вимірювальних функцій за допомогою розкладання до ряду Тейлора. Тому для нелінійних систем втрата фази високого порядку в процесі лінеаризації та створення вихідного результату не є справжнім очікуванням та дисперсією розподілу.

Спеціальна фільтрація UKF має кращу узгодженість у нелінійних системах, ніж спеціальна фільтрація EKF. У той же час спеціальна фільтрація UKF має велику обчислювальну складність – обчислювальний обсяг лінійно зростає з розмірністю.

Інша спеціальна фільтрація популярніша, ніж спеціальні фільтрації EKF і UKF, для застосування в математичній моделі на базі методу SLAM і це є – фільтр частинок PF (рисунок 1.12). Спеціальна фільтрація PF найкраще підходить для аналізу нелінійних систем. Тому для досягнення поставленої мети за допомогою методу SLAM, у якості заміни фільтра Калмана, все частіше застосовують фільтр частинок PF. Однак швидкість обчислень при використанні PF-фільтрів менша, ніж при застосуванні EKF та UKF.

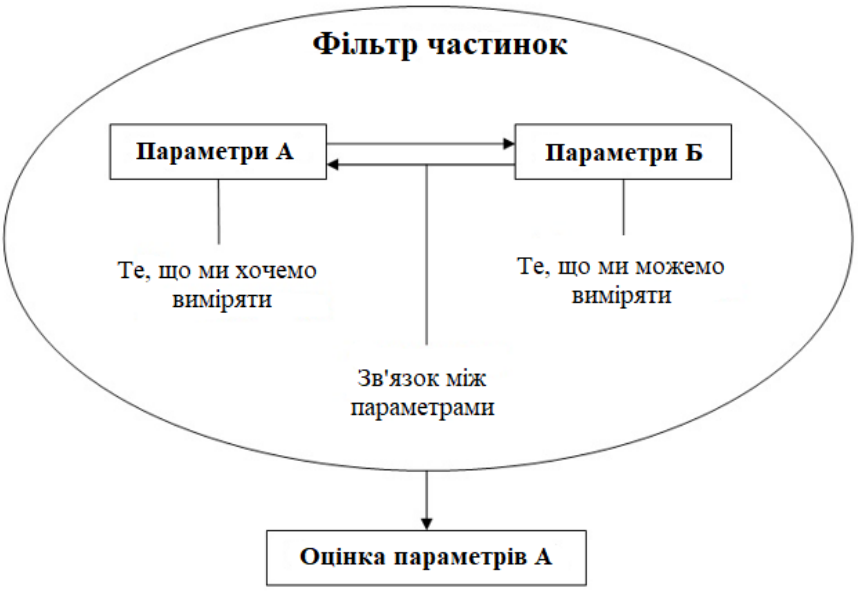


Рисунок 1.12 – Приклад фільтру частинок PF

Так як до позиціонування інтелектуальних мобільних "s-bots", що

входять до складу однієї "Swarm-bot"-system в physical unorganized environment, пред'являються високі вимоги в реальному часі, підвищення швидкості обчислень при використанні PF-фільтрів є актуальним науковим завданням.

1.2.2 Аналіз вирішення задачі SLAM за допомогою відеокамер VSLAM

З розвитком технології обробки зображення почало розвиватися дослідження VSLAM. Наприклад, на рисунку 1.13 представлено функціональну схему роботи алгоритму VSLAM на прикладі домашнього інтелектуального мобільного "s-bots".

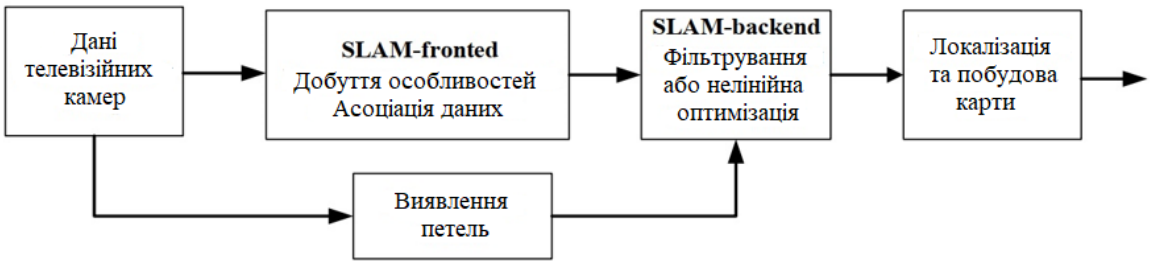


Рисунок 1.13 – Функціональна схема роботи алгоритму VSLAM

Репозиціонуюча здатність лазерного SLAM погана. Лазерний SLAM важко повернути до робочого стану після втрати стеження чи позиціонування. VSLAM може компенсувати цей недолік, але цей алгоритм погано працює в нетекстурованих середовищах, таких як пісок з меншими перешкодами (камені) та в особливо слабких або сильних середовищах освітлення. Порівняно з лазерним SLAM, у VSLAM існує додатковий блок – блок виявлення петель (або замикання петель – Loop closure), який дозволяє зменшити накопичення помилок в процесі реалізації візуального SLAM.

В алгоритмі VSLAM завдання отримання відстежуваного становища інтелектуальних мобільних "s-bots", що входять до складу однієї "Swarm-

bot"-system в physical unorganized environment, здійснюється з накопиченням помилки. У міру того, як шлях продовжує розширюватися, помилка попереднього кадру буде передана безпосередньо в наступному кадрі. Виявлення петель забезпечує зв'язок між поточними та всіма попередніми даними. Тому, коли відстеження втрачено, також можливе використання процедури виявлення петель, щоб зробити репозиціонування (відновлення положення).

Успіх виявлення петель також доводить працездатність алгоритму навігації та позиціонування. У візуальному backend-SLAM, крім використання методу оцінки стану на основі алгоритму фільтрації, тепер також використовується метод на основі нелінійної оптимізації (наприклад, на основі оптимізації граф) для оцінки позиції інтелектуальних мобільних "s-bots", що входять до складу однієї "Swarm-bot"- system в physical unorganized environment, і перешкод.

Однак, візуальний SLAM дає нестабільні результати якщо:

- велика швидкість кутового руху video-камери;
- велика лінійна швидкість руху об'єкта з video-камери;
- немає достатньої кількості характерних точок сцени.

Спільне використання візуальних даних, що надаються інерційною навігаційною системою, є в даний час досить актуальною проблемою досліджень у галузі навігації.

Комбінація (комплексування) цих даних дозволяє забезпечити автономну навігацію для інтелектуальних мобільних "s-bots", що входять до складу однієї "Swarm-bot"-system у physical unorganized environment, наприклад, при вирішенні наступних завдань:

- розвідка аварійних підземних шахт;
- створення карт навколишнього середовища у важкодоступній місцевості, тощо.

Причому даний спосіб комплексування дозволяє зробити систему дешевшою та з меншою вагою обладнання. При комплексуванні систем їх

технічні засоби доповнюють одне одного. Хоча video-камера, як правило, працює повільніше, ніж штучна нейронна мережа (ШНМ), проте методи обробки зображень здатні надати більш точну інформацію про місцезнаходження інтелектуальних мобільних "s-bots", що входять до складу однієї "Swarm-bot"-system в physical unorganized environment. Штучна нейронна мережа в автономному режимі має властивість накопичувати помилки. Тому актуальним є проведення досліджень, які полягають у поєднанні візуального SLAM з результатами оцінки про місцезнаходження інтелектуальних мобільних "s-bots", що входять до складу однієї "Swarm-bot"-system у physical unorganized environment. Завдяки їх об'єднанню, точність оцінки розташування інтелектуальних мобільних "s-bots" підвищується, і вся система стає більш стійкою. У більшості підходів для вирішення задач навігації за допомогою візуально-інерційної системи (ВІС) використовуються алгоритми на основі спеціальної фільтрації або оптимізації графів.

Методи, які побудовані на оптимізації графів, більш точні, але, як правило, мають великі обчислювальні витрати, а алгоритми, що побудовані на спеціальній фільтрації, добре підходять для вирішення завдань у реальному часі. Алгоритм спеціальної фільтрації працює швидше та вимагає менше обчислювальних ресурсів.

Іншою відмітною ознакою алгоритмів візуально-інерційної системи є застосування слабко-пов'язаних або тісно-пов'язаних методів комплексування.

Слабо-пов'язані методи комплексування зазвичай включають два окремі модулі для оцінки стану системи:

- для оцінки стану візуальної інформації;
- для оцінки стану за даними штучної нейронної мережі.

Оскільки в цих методах обробки не враховується взаємозв'язок візуальної та інерційної інформації, то в цьому випадку, наприклад, неможливо виправити помилки у візуальній оцінці, наприклад – алгоритм

EKF.

Тісно-пов'язані методи забезпечують систематичне поєднання (злиття) візуальної інформації та даних, отриманих за допомогою штучної нейронної мережі. Зазвичай ці методи призводять до найкращих результатів. Тому основну увагу в кваліфікаційній роботі приділено дослідженню тісно-пов'язаних методів фільтрації комплексування для розв'язання задачі SLAM, а саме – алгоритму, розробленому на основі методу спеціальної фільтрації Калмана з кількома станами.

1.3 Задача планування маршруту руху інтелектуальних мобільних "s-bots", що входять до складу однієї "Swarm-bot"-system у physical unorganized environment

Після того, як інтелектуальний мобільний "s-bot", що входить до складу однієї "Swarm-bot"-system, за допомогою методу SLAM створив карту physical unorganized environment, розв'язання задачі автономної навігації "s-bot" вимагає формування траєкторії руху з поточного положення до заданої мети.

Планування маршруту руху інтелектуального мобільного "s-bot" у physical unorganized environment - це одна з ключових технологій для забезпечення безпечного обходу перешкод. В даний час відомі такі методи планування маршруту:

- метод штучного потенціального поля;
- алгоритм мурашиної колонії;
- алгоритм імітації відпалу;
- алгоритм A* та його оптимізований варіант.

Після побудови карти фізичного unorganized environment метод SLAM зазвичай створює мережеву карту. Ця мережна карта описує physical unorganized environment через її поділ кількома сітками рівного розміру і встановлює різні характеристики кожної комірки такої сітки:

- схил;
- висота.

Аналізуючи сітку за цими характеристиками, можна спланувати маршрут руху інтелектуального мобільного «s-bot» в обхід перешкод. Завдяки простим обчисленням, метод сітки широко використовується при вирішенні проблем планування маршруту при русі інтелектуального мобільного "s-bot" у physical unorganized environment.

До алгоритмів планування маршруту руху інтелектуального мобільного "s-bot" у physical unorganized environment пред'являються такі вимоги:

- висока швидкість обчислень;
- використання невеликого обсягу пам'яті;
- достовірність отриманих даних.

Враховуючи всі вищеперелічені вимоги, найбільш простим та ефективним алгоритмом у задачі планування маршруту при русі інтелектуального мобільного "s-bot" у physical unorganized environment є алгоритм A* та його модифікації.

У 1968 році група дослідників на чолі з Hart запропонували алгоритм пошуку A*, який є евристичним алгоритмом.

У 1985 році група дослідників очолювана Korf запропонували алгоритм IDA* – Iterative Deepening A* algorithm. У поєднанні з алгоритмом A* та ідеєю ітераційного поглиблення, алгоритм IDA* покращив швидкість обчислення порівняно з алгоритмом A*.

У 1994 році група дослідників, очолювана Stentz, запропонували покращений алгоритм A* у динамічному середовищі – алгоритм D* (Dynamic A* algorithm), який може застосовуватись у задачах для планетоходів. Однак цей алгоритм не передбачає переміщень у різних кутових напрямках. Оскільки на напрямок руху накладається обмеження, то в цьому випадку найкоротший шлях не може бути отриманий.

У 2007 році група дослідників, очолювана Nach, запропонували

алгоритм Basic Theta* – Алгоритм планування траєкторії руху у будь-якому напрямку (АПТРуБН). Цей алгоритм більш простий та швидший. Nach та його група дослідників довели ефективність та надійність роботи алгоритму Basic Theta*. Потім Nach та його група дослідників покращили деякі характеристики та назвали оптимізований алгоритм – Lazy Theta, на основі алгоритму Basic Theta*.

У кваліфікаційній роботі досліджується алгоритм Lazy Theta*, для планування маршруту руху інтелектуальних мобільних "s-bots", що входять до складу однієї "Swarm-bot"-system в physical unorganized environment. Розробники цього алгоритму передбачили можливість при плануванні маршруту враховувати небезпеку рельєфу місцевості, наприклад, під час проведення рятувальних робіт у завалах шахт. Основною проблемою при русі в шахті для колісного інтелектуального мобільного "s-bot" є нахили та перешкоди. При збільшенні кута нахилу поверхні в шахті більше допустимого критичного значення, можливе перекидання "s-bos". Подолання перешкод – це одне з основних завдань, яке має вирішувати кожен інтелектуальний мобільний "s- bot", що входить до складу однієї "Swarm-bot"-system в physical unorganized environment, для зміни свого розташування у шахті. При русі інтелектуального мобільного "s-bot" по поверхні шахти може виникнути ситуація, коли колеса "s-bot" можуть застрягти в ямах, у западинах та інших нерівностях рельєфу. Тому перш ніж визначати показники небезпеки місцевості, бажано провести аналіз рельєфу physical unorganized environment, у якій передбачається працювати "s-bot" і вирішувати завдання локалізації.

2 АЛГОРИТМ LAZY THETA* ПЛАНУВАННЯ МАРШРУТУ РУХУ ІНТЕЛЕКТУАЛЬНОГО МОБІЛЬНИЙ "S-BOT" З УРАХУВАННЯМ РЕЛЬЄФУ МІСЦЕВОСТІ

Одне з найважливіших завдань, що вимагає вирішення при плануванні маршруту руху інтелектуального мобільного "s-bot", що входить до складу однієї "Swarm-bot"-system у physical unorganized environment, полягає в тому, щоб його траєкторія руху від початкової до кінцевої (цільової) точки маршруту в загальному випадку, що проходить у маловивченому середовищі на поверхні шахти, забезпечувала його гарантовану безпеку. Тому розробка алгоритму планування маршруту інтелектуального мобільного "s-bot", який в автономному режимі забезпечував би пошук оптимальної та безпечної траєкторії, є в даний час дуже актуальним завданням.

На карті місцевості, створеної на основі інформації, отриманої від робітників у шахті, як правило, відсутня точна інформація про локальний розподіл невеликих перешкод (наприклад, каменів). Тому такі карти не можуть бути достатньо ефективно використані при плануванні маршруту. Інформація, яка міститься в карті, отриманої в результаті застосування методу SLAM, є більш точною. Тому завдання планування маршруту може бути виконано з використанням цієї карти.

Загальна структура методу SLAM із завданням планування маршруту інтелектуального мобільного "s-bot" показана в першому розділі на рисунку 1.10.

Як описано у першому розділі, для зменшення протяжності маршруту, врахування реального рельєфу та підвищення швидкості обчислень, необхідно розробити та дослідити відповідні модифікації алгоритмів планування траєкторії руху інтелектуального мобільного "s-bot" у будь-якому напрямку на основі алгоритму A*.

2.1 Огляд алгоритмів пошуку маршруту руху інтелектуального мобільного "s-bot" у physical unorganized environment

2.1.1 Огляд алгоритму A*

Алгоритми A* є евристичними алгоритмами. Евристичний алгоритм (евристика) – це алгоритм розв'язання задачі, що включає практичний метод, що не є гарантовано точним чи оптимальним, але достатній для вирішення поставленого завдання (рисунок 2.1).

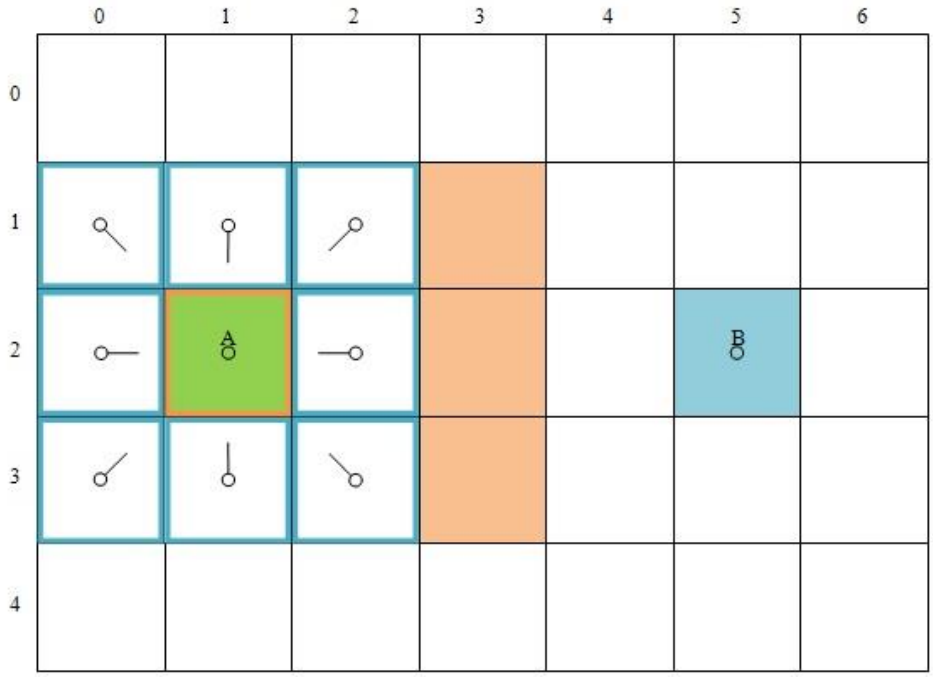


Рисунок 2.1 – Приклад знаходження маршруту за допомогою алгоритму A*

Простіше кажучи, евристика – це не повністю математично обґрунтований чи навіть іноді не зовсім коректний, але при цьому практично корисний алгоритм. Його переваги полягають у високій ефективності пошуку, простоті реалізації та можливості прискореного вирішення завдання у тих випадках, коли точне рішення не може бути знайдено.

Алгоритм A* має багато цікавих властивостей. Він гарантовано знаходить найкоротший шлях до того часу, поки евристичне наближення є допустимим, тобто ніколи не перевищує дійсної відстані, що залишилася, до мети. На відміну від інших, цей алгоритм найкраще використовує евристику, обробляючи найменшу кількість вузлів. Щоб скористатися цим алгоритмом, необхідно виконати перетворення карти на граф. Для карти-сітки таке відображення є стандартним, при цьому карта є восьмизв'язковим графом, вершини якого знаходяться в центрі кожної комірки, а ребрами – є переходи до сусідніх. Кожна комірка може перебувати в стані:

- «вільно»;
- «зайнято».

Карта сіткових функцій має ряд переваг, зокрема:

- відома відстань від кожної комірки до найближчої перешкоди;
- враховуються габарити інтелектуального мобільного "s-bot";
- відстань до перешкоди використовується в якості вартості переходу.

За допомогою евристичних алгоритмів пошуку можна знаходити маршрути руху від однієї вершини (початкової) до іншої вершини (цільової, кінцевої) з найменшими витратами (вартістю). Порядок обходу вершин під час руху інтелектуального мобільного "s-bot" за маршрутом визначається евристичною функцією, яка враховує показники «відстань + вартість» і має такий вигляд:

$$f(n) = g(n) + h(n), \qquad (2.1)$$

де f(n) – значення евристичної функції;

n – позначення (номер) розглянутої вершини;

g(n) – значення функції вартості досягнення розглянутої вершини з початкової вершини (може бути евристичною функцією, або не евристичною);

h(n) – значення функції евристичної оцінки відстані від розглянутої

(поточної) вершини до кінцевої (має бути допустимою евристичною оцінкою).

У загальному випадку функцію h(n) можна обчислити різними способами. У кваліфікаційній роботі використовується один із найпоширеніших методів оцінки відстані, а саме:

$$h(n)=\sqrt{(x_t-x_s)^2+(y_t-y_s)^2}, \qquad (2.2)$$

де x_t , x_s – значення абсцис поточної та цільової вершини;

y_t, y_s – значення ординат поточної та цільової вершини.

Розглянемо властивості та особливості кількох алгоритмів пошуку маршрутів руху.

Алгоритм А* – це алгоритм пошуку, який дозволяє знайти у виваженому графі маршрут із найменшим значенням евристичної функції f (n) при русі від початкової вершини до заданої кінцевої вершини.

Алгоритм А* можна спростити так, як показано на рисунках 2.2 - 2.5.

На рисунках наведено такі позначення:

- s – розглянута вершина;
- parent (s) – батьківська вершина;
- s' – розширена вершина;
- g (parent(s)) – значення функції вартості шляху досягнення батьківської вершини parent(s) від початкової вершини;
- b (parent(s), s') – відстань від батьківської вершини parent(s) до розширеної вершини s';
- Neighbour(s) – сусідні вершини, які можна бачити.

Розглянемо основні кроки виконання алгоритму А*.

Крок № 1. Точка s_0 (рисунок 2.2) є початковою вершиною, а також розглянутою вершиною.

Точки s'_0 (Neighbour(s_0)) є її сусідніми вершинами розширення, як

показано на рисунку 2.2. Вирахуємо всі значення евристичної функції f (s_0) для початкової (розглянутої) вершини s_0.

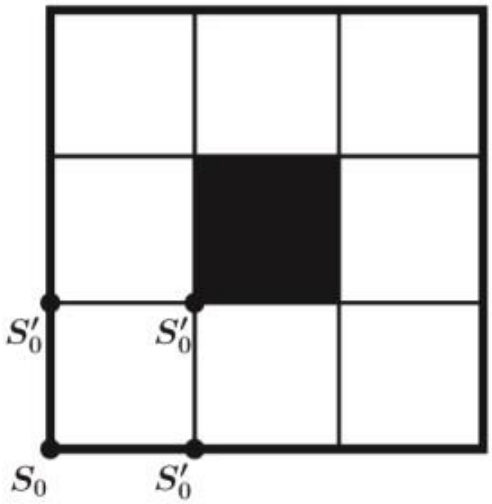


Рисунок 2.2 – Приклад роботи спрощеного алгоритму A*

Крок № 2. Виберемо вершину розширення s', відповідну найменшому значенню функції f (s_0). Як вершина, що розглядається, приймається вершина s_1 (тобто $s'_0 = s_1$). Потім оновлюються шляхи так, як показано на рисунку 2.3.

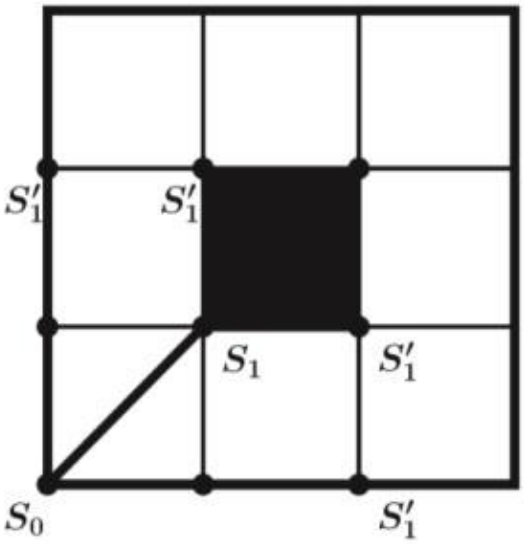


Рисунок 2.3 – Приклад роботи спрощеного алгоритму A*

Розширивши вершину s_1 у різних напрямках, отримаємо вершини розширення s'_1 (Neighbour(s_1) та s'_1 (вершини не були розширені на минулому етапі). Обчислимо всі значення евристичної функції $f(s_1)$ для розглянутої вершини s_1.

Крок № 3. Виберемо вершину розширення s'_1, відповідну найменшому значенню $f(s_1)$. Далі, в якості розглянутої вершини приймається вершина s_2 (тобто $s'_1 = s_2$). Оновимо шляхи так, як показано на рисунку 2.4.

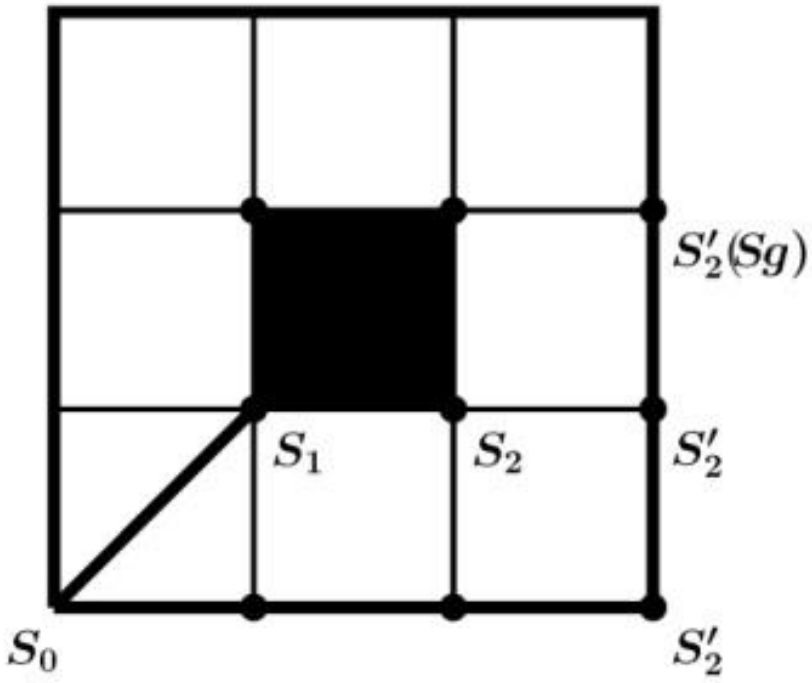


Рисунок 2.4 – Приклад роботи спрощеного алгоритму A*

Розширивши вершину s_2, отримаємо вершини розширення s'_2 (Neighbour(s_2) та s'_2 (вершини не були розширені на минулому етапі) так, як показано на рисунку 2.4.

Крок № 4. Цільова вершина s_g знаходиться в одній із вершин розширення s'_2. Оновимо шляхи, як показано на рисунку 2.5. Таким чином,

підсумковий шлях від початкової до кінцевої точки маршруту s_0 - s_1 - s_2 - s_g показаний на рисунку 2.5.

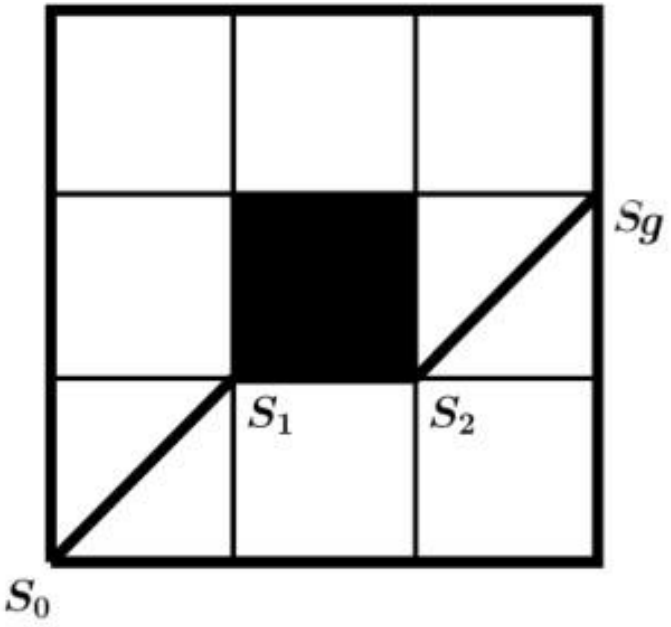


Рисунок 2.5 – Приклад роботи спрощеного алгоритму A*

2.1.2 Огляд Алгоритму планування руху в будь-якому напрямку

Алгоритм планування траєкторії руху у будь-якому напрямку (АПТРуБН) заснований на алгоритмі A*. За допомогою даного алгоритму, порівняно з алгоритмами A* та D*, можливо знайти більш короткі маршрути руху інтелектуального мобільного "s-bot", що входить до складу однієї "Swarm-bot"-system у physical unorganized environment. Це є наслідком того, що в алгоритмах A*, D* рух інтелектуального мобільного "s-bot" формується, головним чином, або вздовж кордонів комірки сітки, або по її діагоналі.

Алгоритм планування траєкторії руху у будь-якому напрямку, у порівнянні з алгоритмами Basic Theta* та Lazy Theta*, не має обмежень у виборі напрямку руху. Наприклад, як видно з рисунку 2.6 та рисунку 2.7, на траєкторії, реалізованої за допомогою алгоритму планування траєкторія руху

в будь-якому напрямку інтелектуального мобільного "s-bot", вдається уникнути зайвих змін (відхилень, вигинів) під час руху у напрямку від початкової точки A4 до цільової точки C1 маршруту у physical unorganized environment.

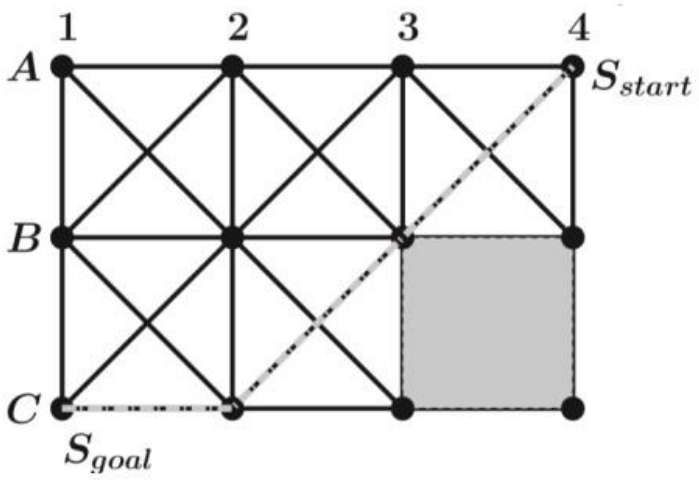


Рисунок 2.6 – Приклади траєкторії руху при фіксованому куті

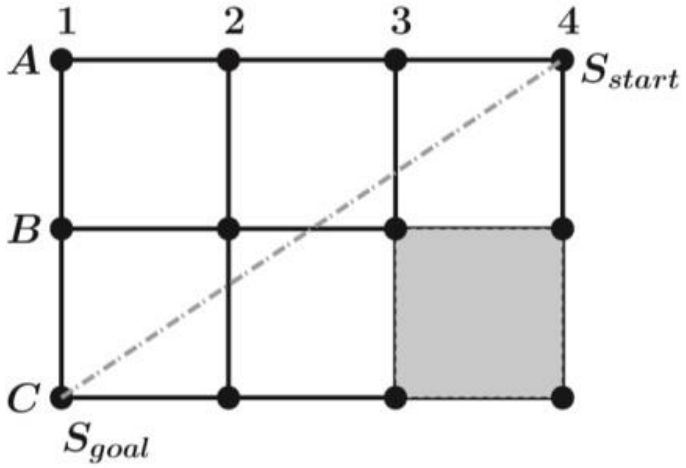


Рисунок 2.7 – Приклади траєкторії руху у будь-якому напрямку

Тому за допомогою алгоритму планування траєкторії руху в будь-якому напрямку інтелектуального мобільного "s-bot" можна значною мірою

скоротити результуючий шлях. Таким чином, вивчення властивостей та особливостей даного алгоритму становить особливий інтерес. В алгоритмах планування траєкторії руху в будь-якому напрямку інтелектуального мобільного "s-bot" траєкторії руху можуть оновлюватись згідно з двома варіантами.

Варіант № 1 (рисунок 2.8) – шлях оновлюється між поточною вершиною та сусідньою вершиною розширення.

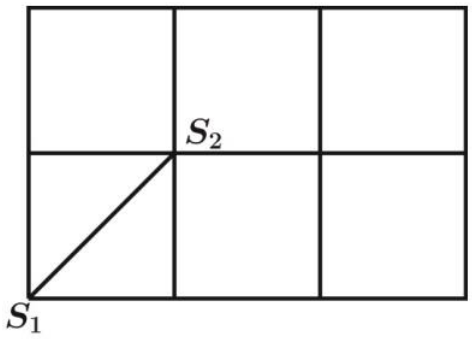


Рисунок 2.8 – Оновлення шляху руху інтелектуального мобільного "s- bot"

Варіант № 2 (рисунок 2.9) – шлях оновлюється між батьківською вершиною (parent) та довільною вершиною розширення.

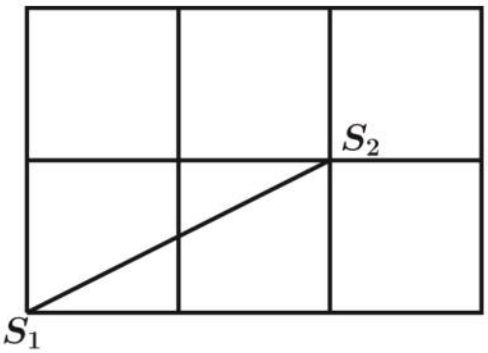


Рисунок 2.9 – Оновлення шляху руху інтелектуального мобільного "s- bot"

2.1.3 Огляд алгоритму Basic Theta*

Алгоритм Basic Theta* є модифікацією алгоритму A* і має певну схожість з алгоритмом D*. Відмінна особливість алгоритму Basic Theta* – здатність забезпечувати рух інтелектуального мобільного "s-bot", що входить до складу однієї "Swarm-bot"-system у physical unorganized environment, коротким шляхом з меншою кількістю змін у напрямку руху. При цьому траєкторія може поширюватися у будь-якому напрямку, не обов'язково через найближчі вершини, але за умови, що між ними є пряма видимість.

Основні кроки алгоритму Basic Theta* показані на рисунках 2.10 - 2.15.

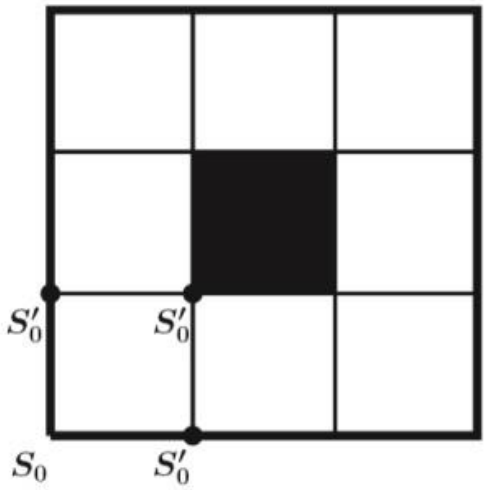


Рисунок 2.10 – Крок № 1 алгоритму Basic Theta*

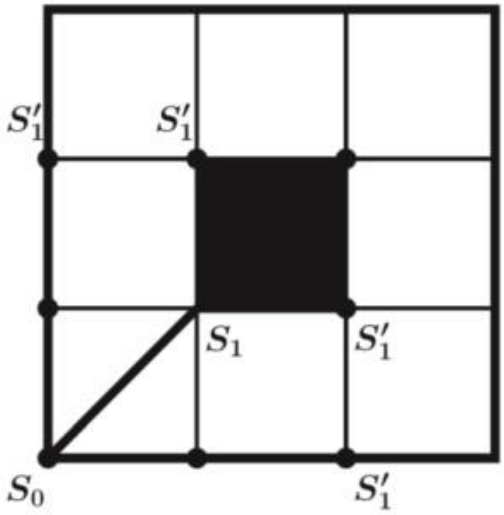


Рисунок 2.11 – Крок № 2 алгоритму Basic Theta*

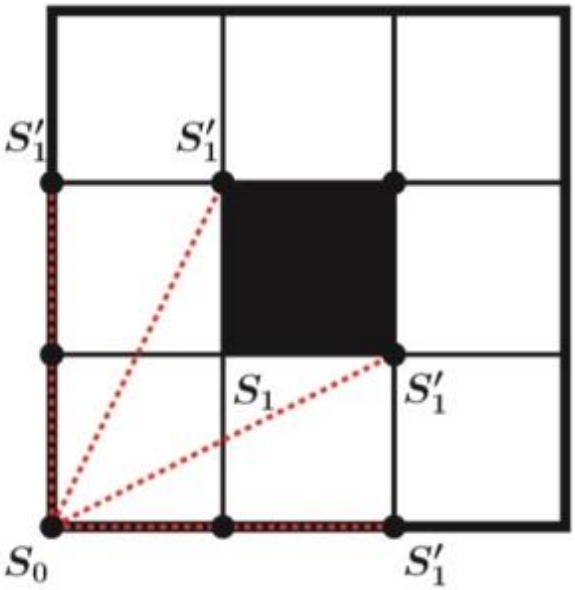


Рисунок 2.12 – Крок № 3 алгоритму Basic Theta*

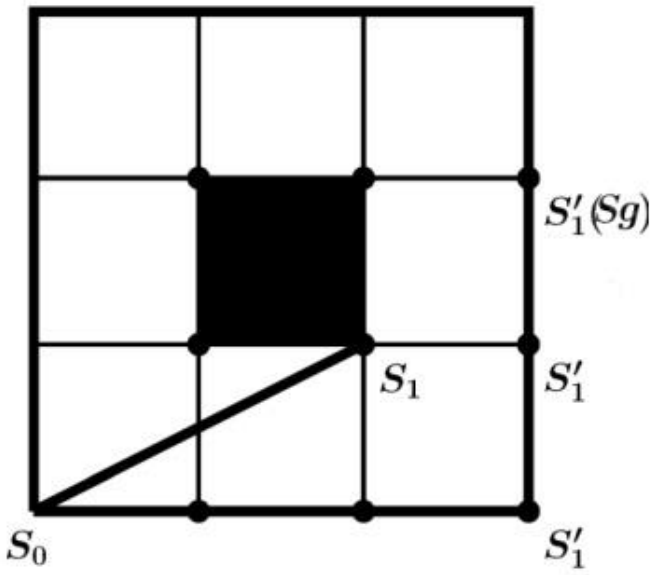


Рисунок 2.13 – Крок № 4 алгоритму Basic Theta*

Проаналізуємо основні етапи алгоритму Basic Theta*:

- крок № 1 – зміст етапу аналогічний крок № 1 алгоритму A* (рисунок 2.10);
- крок № 2 – зміст етапу аналогічний крок № 2 алгоритму A* (рисунок 2.11);
- крок № 3 – перевірка поля зору (пунктирна червона лінія на рисунку 2.12 між батьківською вершиною (parent(s_1) = s_0) і всіма вершинами розширення s'_1). Якщо між батьківською вершиною (parent(s_1) = s_0) та вершиною розширення s'_1 є пряма видимість, то значення евристичних функцій $f(s_1) = g(parent(s_1)) = b(parent(s_1), s'_1) = h(s'_1, s_g)$. Якщо між батьківською вершиною (parent(s_1) = s_0) та вершиною розширення s'_1 немає прямої видимості, то значення евристичних функцій $f(s_1) = g(s_1) = b(s_1, s'_1) = h(s'_1, s_g)$. Вибір вершини розширення s'_1 відповідає найменшому значенню евристичних функцій $f(s_1)$ в якості розглянутої вершини s_1. На рисунку 2.12 показаний приклад, коли між батьківською вершиною та відповідною вершиною розширення s'_1 є пряма видимість;
- крок № 4 – згідно з варіантом № 2 оновлення шляху приймаємо

(parent(s_1) = s_0) та s'_1 = s_1. Розширюємо вершину s_1 (рисунок 2.13) та отримуємо вершини розширення s'_1, які не були розширені на попередньому етапі (Neighbour(s_1) та s'_1);

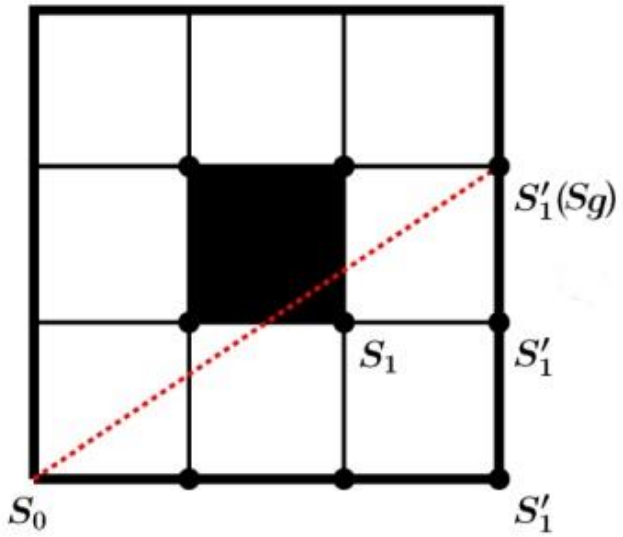


Рисунок 2.14 – Крок № 5 алгоритму Basic Theta*

- крок № 5 – цільова вершина s_g знаходиться в одній із вершин розширення s'_1. Якщо між батьківською вершиною parent(s_1) = s_0 та цільовою вершиною s_g немає прямої видимості (на рисунку 2.14 червона пунктирна лінія), то подальше оновлення шляху між поточною (розглянутою) вершиною s_1 та цільовою вершиною s_g виконується згідно з варіантом № 1;

- крок № 6 – на цьому робота алгоритму руху інтелектуального мобільного "s-bot", що входить до складу однієї "Swarm-bot"-system у physical unorganized environment, закінчується. Підсумковий шлях від початкової вершини до цільової вершини s_0 - s_1 - s_g показаний на рисунку 2.15.

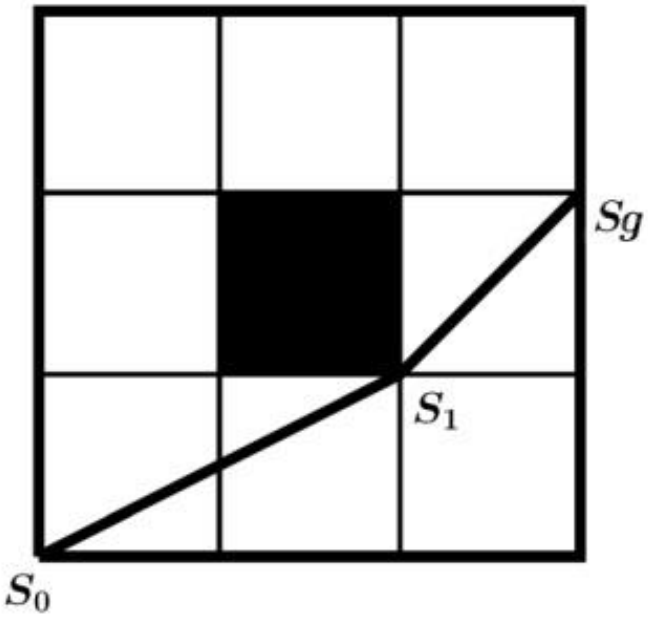


Рисунок 2.15 – Крок № 6 алгоритму Basic Theta*

2.2 Дослідження алгоритму Lazy Theta*

Алгоритм Lazy Theta* представляє собою модифікацію алгоритму Basic Theta*, який забезпечує більш високу швидкість обчислень, оскільки в даному алгоритмі реалізований простий підхід до виконання перевірки видимості вершин розширення, що знаходяться в даний час у зору з батьківської вершини. Алгоритм Lazy Theta* – це новий алгоритм, розроблений на основі алгоритму Basic Theta* в результаті оптимізації та комбінування переваг попередніх алгоритмів. Даний алгоритм менш складний за структурою і має більш високу швидкодію. Алгоритм Lazy Theta* відрізняється від розглянутих алгоритмів процедурою пошуку розглянутої вершини, яка дозволяє зменшити час обчислень. В алгоритмі Lazy Theta* ще додається перевірка поля зору між батьківською вершиною parent(parent(s)) та відповідною вершиною розширення s', щоб отримати більш короткий шлях маршруту. Основні кроки алгоритму Lazy Theta*

представлені на рисунках 2.16 - 2.18.

Проаналізуємо основні кроки алгоритму Lazy Theta*:

- крок № 1 – зміст етапу аналогічний кроку № 1 алгоритма A* (рисунок 2.16);

- крок № 2 – зміст етапу аналогічний кроку № 2 алгоритма A* (рисунок 2.17);

- крок № 3 – для алгоритму Lazy Theta* (рисунок 2.18) заздалегідь передбачається, що між батьківською вершиною parent(s_1) = s_0 та вершиною розширення s'_1 має бути пряма видимість.

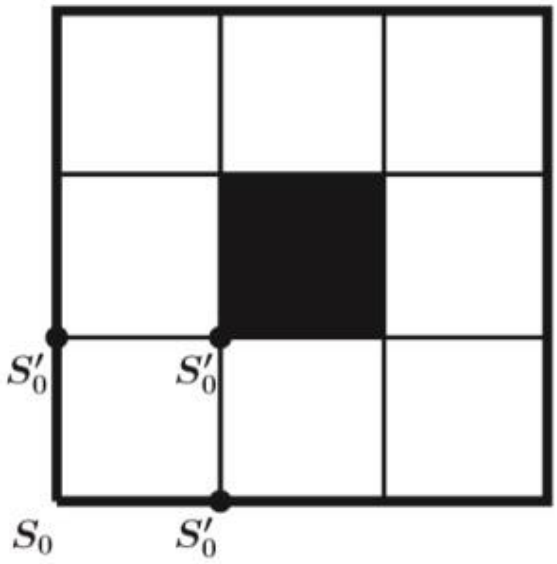


Рисунок 2.16 – Крок № 1 алгоритму Lazy Theta*

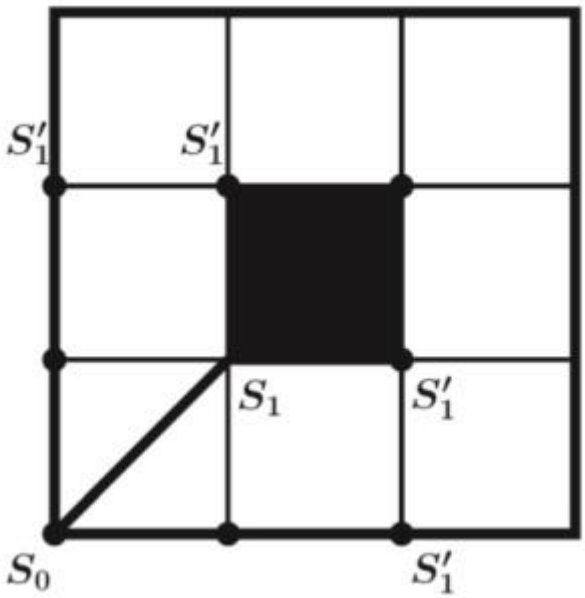


Рисунок 2.17 – Крок № 2 алгоритму Lazy Theta*

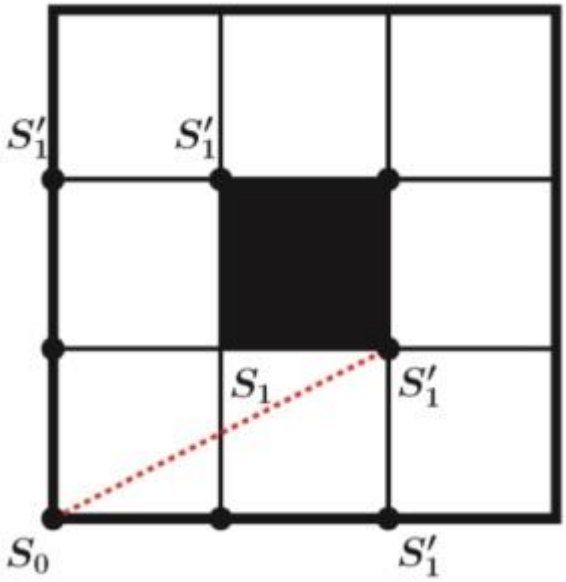


Рисунок 2.18 – Крок № 3 алгоритму Lazy Theta*

Якщо умова $g(parent(s_1)) = b(parent(s_1), s') < g(s'_1)$ виконується, то значення евристичних функцій: $f(s_1) = g(parent(s_1)) + b(parent(s_1), s'_1) + h(s'_1, s_g)$. Якщо умова $g(parent(s_1)) + b(parent(s_1), s') < g(s'_1)$ не виконується, то значення евристичних функцій: $f(s_1) = g(s_1) + b(s_1, s'_1) + h(s'_1, s_g)$.

Виберемо вершину розширення s'_1, що відповідає найменшому значенню евристичних функцій $f(s_1)$ в якості розглянутої вершини s_1. Перевіряється властивість прямої видимості (рисунок 2.18) між батьківською вершиною ($parent(s_1) = s_0$) та відповідною вершиною розширення s'_1 (перевірка виконується лише один раз). На рисунку 2.18 червоною пунктирною лінією показано, що між даними вершинами є пряма видимість, подальше оновлення шляху між батьківською вершиною $parent(s_1)$ та відповідною вершиною розширення s'_1 виконується згідно з варіантом № 2;

- крок № 4 – згідно з варіантом № 2 оновлення шляху приймаємо $parent(s_1) = s_0$ и $s'_0 = s_1$. Розширюємо вершину s_1 (рисунок 2.19) і отримуємо вершини розширення s'_1, які не були розширені на попередньому етапі ($Neighbour(s_1)$ та s'_1);

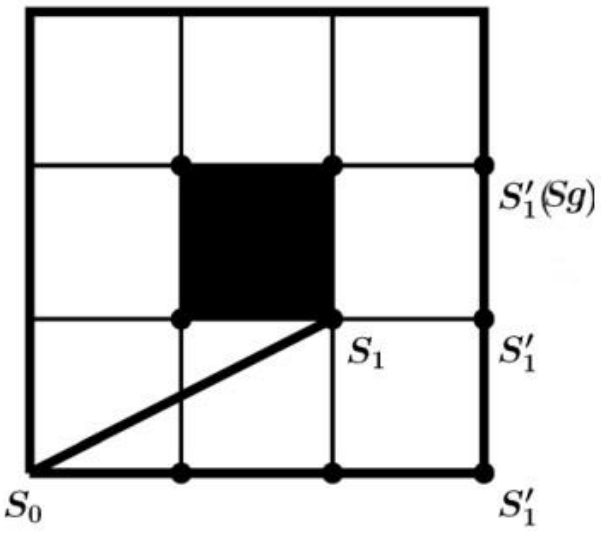


Рисунок 2.19 – Крок № 4 алгоритму Lazy Theta*

- крок № 5 – цільова вершина s_g знаходиться в одній із вершин розширення s'_1. Якщо між батьківською вершиною $parent(s_1) = s_0$ та цільовою вершиною s_g немає прямої видимості (на рисунку 2.20 червона пунктирна лінія), то подальше оновлення шляху між поточною (розглянутою) вершиною s_1 та цільовою вершиною s_g виконується згідно з варіантом № 1;

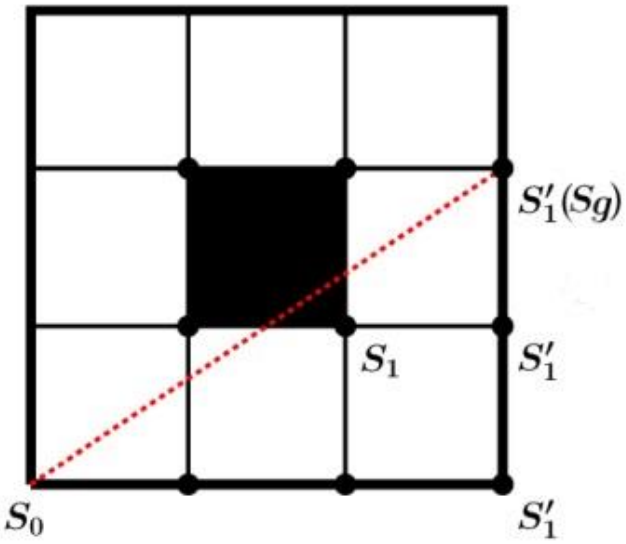


Рисунок 2.20 – Шаг № 5 алгоритма Lazy Theta*

- крок № 6 – на цьому робота алгоритму руху інтелектуального мобільного "s-bot", що входить до складу однієї "Swarm-bot"-system в physical unorganized environment, закінчується. Підсумковий шлях від початкової вершини до цільової вершини s_0 - s_1 - s_g показаний на рисунку 2.21.

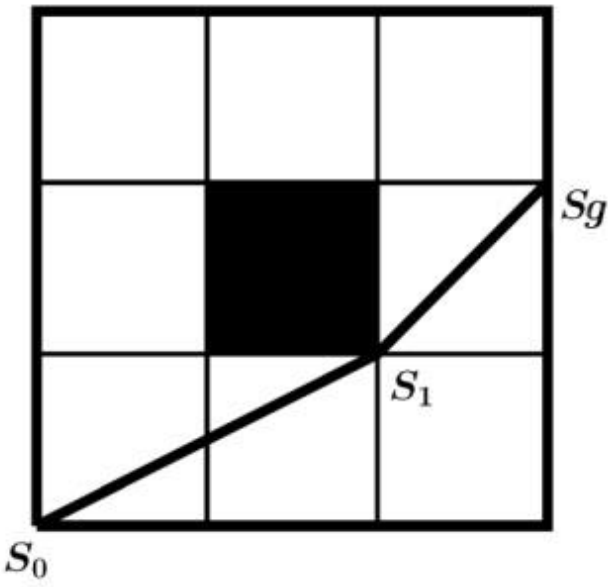


Рисунок 2.21 – Шаг № 6 алгоритма Lazy Theta*

Під час руху інтелектуального мобільного "s-bot", що входить до складу однієї "Swarm-bot"-system в physical unorganized environment, особливо важливо дотримуватися правил безпеки його пересування. На основі аналізу поверхневих особливостей рельєфу в шахті можна судити про показник небезпеки її рельєфу. Показник небезпеки рельєфу є одним із параметрів алгоритму, що оптимізуються. Аналіз показника небезпеки рельєфу допомагає знайти для інтелектуального мобільного "s-bot" траєкторію з низьким рівнем ризику.

3 ДОСЛІДЖЕННЯ АЛГОРИТМУ LAZY THETA* РУХУ "S-BOT" З УРАХУВАННЯМ РЕЛЬЄФУ МІСЦЕВОСТІ

В даний час може використовуватися колісна конфігурація інтелектуального мобільного "s-bot", що входить до складу однієї "Swarm-bot"-system в physical unorganized environment. Основною проблемою під час руху по поверхні в шахті для колісного інтелектуального мобільного "s-bot" є нахили та перешкоди. При величині кута нахилу поверхні у шахті більше його критичного значення можливе перекидання "s-bot". Подолання перешкод – це одна з основних причин для суттєвої зміни положення "s-bot". Ця ситуація часто може виникати на поверхні в шахті та призвести до неузгодженості між колесами "s-bot", а також до застрягання в падинах та інших нерівностях рельєфу. Перш ніж визначити показники небезпеки місцевості, необхідно провести аналіз physical unorganized environment. Виходячи з вимог стійкості положення інтелектуального мобільного "s-bot" при його русі та подоланні перешкод, були обрані такі топографічні коефіцієнти:

- кут нахилу місцевості або крутість схилу;
- шорсткість поверхні;
- розмах рельєфу.

Кут нахилу місцевості або крутість схилу (ската) – кут, що утворюється напрямом схилу з горизонтальною площиною і виражається в кутових мірах або схилах. Аналіз кута нахилу місцевості одна із найважливіших характеристик місцевості. Найбільше значення кута нахилу місцевості, у якому "s-bot" може виконати підйом, становить 30^0 градусів. Тому, для оцінки здатності "s-bot" долати схили, у кваліфікаційній роботі було використано саме це критичне значення даного показника.

Шорсткість поверхні є відображенням ступеня підйому та падіння рельєфу. Шорсткість може впливати на стійкість руху інтелектуального

мобільного "s-bot", що входить до складу однієї "Swarm-bot"-system в physical unorganized environment, що важливо задля забезпечення ефективної роботи його устаткування. Шорсткість рельєфу зазвичай визначається як відношення площі поверхні до його проекції на горизонтальну площину..

Розмах рельефу – різниця абсолютних висот місцевості або відносне перевищення максимальних висот над мінімальними за напрямом руху "s-bot". Цей показник відображає здатність інтелектуального мобільного "s-bot" до подолання перешкод. У кваліфікаційній роботі за найбільшу висоту перешкоди, при якій "s-bot" може виконати підйом, взято параметр у 20 см.

Ґрунтуючись на наведеній вище теорії, у кваліфікаційній роботі досліджується алгоритм Lazy Theta*, у якому враховується чинник загрози рельєфу місцевості.

Таким чином, нова евристична функція набуває наступного вигляду:

$$f(n) = g(n) + h(n) + c(n), \tag{3.1}$$

де c(n) – показник небезпеки для маршруту.

Показник небезпеки руху інтелектуального мобільного "s-bot" за маршрутом від початкової точки s_0 до цільової точки s_g визначається наступним чином:

$$c(s_0, s_g) = \sum_{s_0}^{s_g} \alpha \cdot R(i, j), \tag{3.2}$$

де s_0 – початкова точка маршруту;

s_g – кінцева точка маршруту;

R(i, j) – показник небезпеки рельєфу місцевості на координатній сітці (i, j);

α – відповідний ваговий коефіцієнт.

Якщо довжина шляху через комірку сітки менша ніж $\frac{\sqrt{2}}{2e}$, то ваговий показник α = 0,5. Якщо довжина шляху через сітку більша ніж $\frac{\sqrt{2}}{2e}$, то ваговий показник α = 1. Тут e – значення кроку сітки.

Обчислювальний експеримент проводився за допомогою імітаційного моделювання в середовищі моделювання MATLAB, що обумовлено його функціональністю.

Наприклад, для маршруту, показаного на рисунку 3.1, показник небезпеки має такий вигляд:

$$c(s_0,s_g)=R(1,1)+\frac{1}{2}(R(2,1)+R(2,2))+R(3,2)+R(4,3).$$

На рисунку 3.1 представлені результати моделювання в середовищі MATLAB небезпеки руху інтелектуального мобільного s-bot за заданим маршрутом. При проведенні імітаційного моделювання у середовищі MATLAB порівнювалися алгоритми: A*, Basic Theta*, Lazy Theta*. Результати моделювання алгоритмів для розмірів області на координатній сітці 50×50м^2 наведені у таблиці 3.1. Імітаційне моделювання в середовищі MATLAB проводилося за наступних умов:

- крок координатної сітки дорівнює 1 м;
- кількість випадкових перешкод в області – 10%, 20%, 30%;
- L(м) – довжина шляху;
- t(с) – час обчислення.

Результати імітаційного моделювання в середовищі MATLAB показують, що шлях, отриманий алгоритмом Lazy Theta*, є найкоротшим із усіх. Хоча час обчислення для даного алгоритму більший, ніж для алгоритму A*, однак, порівняно з часом обчислення за допомогою алгоритму планування руху в будь-якому напрямку Basic Theta* – менше. При

збільшенні кількості перешкод час роботи алгоритму Lazy Theta* помітно менше, ніж у решти алгоритмів, що ще раз доводить ефективність даного алгоритму.

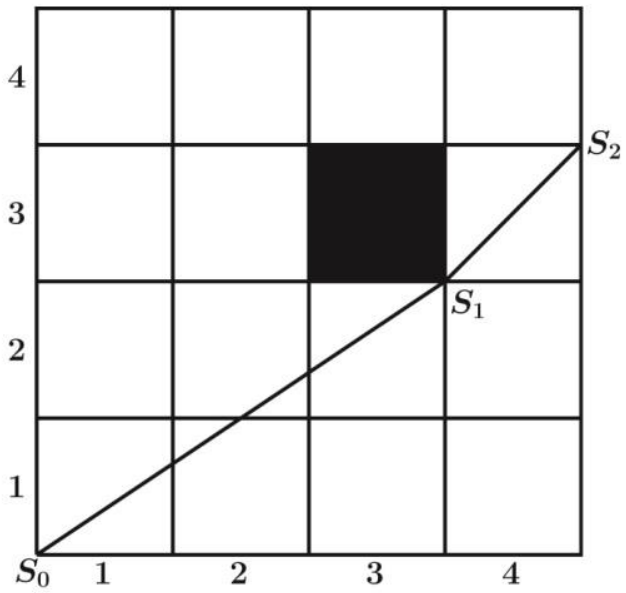


Рисунок 3.1 – Небезпека руху інтелектуального мобільного "s-bot" по заданому маршруту

Таблиця 3.1 – Розмір області на координатній сітці 50×50 м

Кількість випадкових перешкод	A*		Basic Theta*		Lazy Theta*	
	L(м.)	t(с.)	L(м.)	t(с.)	L(м.)	t(с.)
10%	70.486	0.207	70.205	0.305	70.202	0.304
20%	72.801	0.208	72.206	0.307	72.203	0.306
30%	75.307	0.209	76.207	0.309	76.204	0.308

Результати роботи алгоритмів Basic Theta* та Lazy Theta* з урахуванням перешкод, нанесених на координатну сітку, показано на рисунку 3.2.

На рисунку 3.2 позначено:

- чорне коло – реальні перешкоди;

- червоні кола – межі віртуальних перешкод (збільшення розмірів перешкод необхідно для їх гарантованого обходу та запобігання зіткненню);

- синє коло – розширена визначена точка (s);

- зелена безперервна лінія – шлях, побудований за допомогою алгоритму Basic Theta*;

- червона безперервна лінія – шлях, побудований за допомогою алгоритму Lazy Theta*.

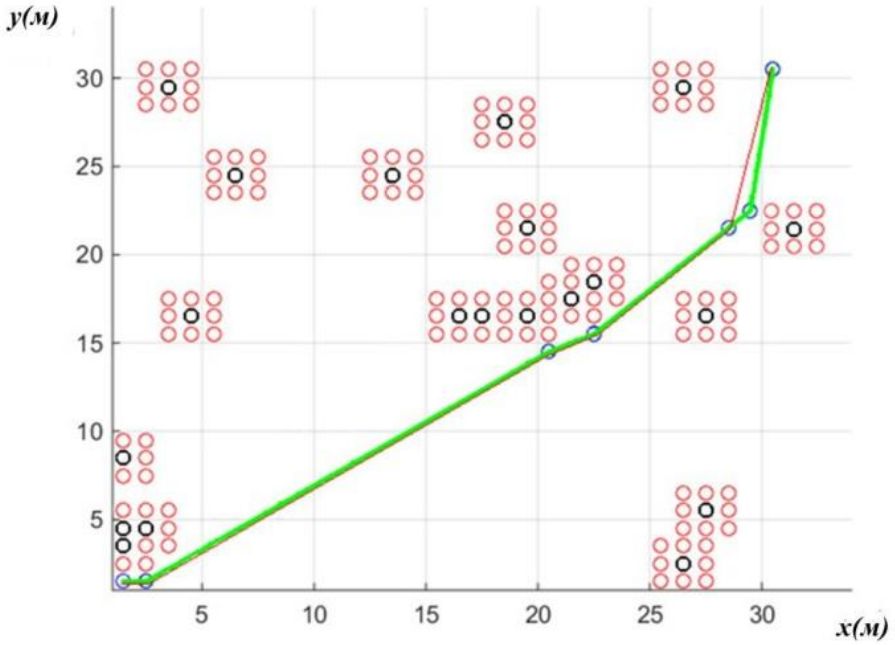


Рисунок 3.2 – Порівняльний аналіз роботи алгоритмів Basic Theta* та Lazy Theta*

ВИСНОВКИ

У кваліфікаційній роботі було досліджено алгоритм Lazy Theta* та його застосування в системах навігації та управління рухом інтелектуальним мобільним "s-bot", що входить до складу однієї "Swarm-bot"-system у physical unorganized environment.

Проведено дослідження актуального науково-технічного завдання, пов'язаного з автономною системою управління інтелектуальним мобільним "s-bot" з можливістю планування маршруту за допомогою алгоритму Lazy Theta* у складному середовищі з урахуванням фактора небезпеки рельєфу місцевості. В даний час одна з найважливіших проблем, що вимагає вирішення при плануванні маршруту руху інтелектуального мобільного "s-bot", полягає в тому, що його траєкторія руху від початкової до кінцевої (цільової) точки маршруту в загальному випадку проходить в маловивченій обстановці.

Цей алгоритм у режимі реального часу зможе забезпечувати пошук короткої та безпечної траєкторії.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Ткачов В.М., Токарєв В.В., Радченко В.О., Лебедєв В.О. Проблема передачі даних типу BIG DATA у мобільній системі «Мультикоптер-сенсорна мережа» / В.М. Ткачов, В.В. Токарєв, В.О. Радченко, В.О. Лебедєв // Системи управління, навігації та зв'язку. - 2017. №2(42). - сс.154-157

2. Пат. 118921 Україна, МПК H04W 64/00. Спосіб передачі цифрових даних мультикоптерною системою між сегментами розподіленої сенсорної мережі та базовою станцією / В.М. Ткачов, В.В. Токарєв - № u201704085; заявл. 24.04.2017; опубл. 28.08.2017. Бюл. № 16. 5с.

3. Радченко В.А., Токарєв В.В., Ткачев В.М. Мобільна система передачі даних на базі динамічно реконфігурованих мультикоптерних пристроїв / В.О. Радченко, В.В. Токарєв, В.М. Ткачов // Проблеми інформатизації: тези доповідей V - наук. - техн. конф., 13 - 15 лист. 2017р. - Харків, 2017. - сс.36.

4. Створення науково-методичних основ забезпечення живучості мережевих систем обміну інформацією в умовах зовнішнього впливу потужного НВЧ випромінювання: звіт про НДР (заключ.) № держреєстрації 0117U003916.: Ф76/109-2017 / Харків. нац. ун-т радіоелектроніки; керівник Г. И. Чурюмов. – Харків, 2017. – 116 с.

5. Ruban I.V., Churyumov G.I., Tokarev V.V., Tkachov V.M. Provision of Survivability of Reconfigurable Mobile System on Exposure to High-Power Electromagnetic Radiation / I.V. Ruban, G.I. Churyumov, V.V. Tokarev, V.M. Tkachov // Selected Papers of the XVII International Scientific and Practical Conference on Information Technologies and Security: (ITS 2017). CEUR Workshop Processing., 30 nov. 2017 y. - Kyiv, 2017. - pp. 105-111.

6. Serkov A., Kravets V., Yakovenko I., Churyumov G., Tokariev V., Nannan W. Ultra Wideband Signals in Control Systems of Unmanned Aerial Vehicles / A. Serkov, V. Kravets, I. Yakovenko, G. Churyumov, V. Tokariev, W.

Nannan // The 10h IEEE International Conference on Dependable Systems, Services and Technologies: (DESSERT'2019)., 5-7 june, 2019 y. - Leeds, 2019. - pp.26 - 29.

7. Серков О. А., Пустовойтов П. Є., Яковенко І. В., Лазуренко Б. О., Чурюмов Г. І., Токарєв В. В., Наннан Ванг. Надширокосмугові технології в системах управління мобільними об'єктами. / О. А. Серков, П. Є. Пустовойтов, І. В. Яковенко, Б. О. Лазуренко, Г. І. Чурюмов, В. В. Токарєв, Ванг Наннан // Сучасні інформаційні системи. - 2019. - Т.3, №2. - сс.22-27.

8. Серков О.А., Князєв В.В., Лазуренко Б.О., Яковенко І.В., Чурюмов Г.І., Токарєв В.В. Надширокосмугові технології в задачах забезпечення електромагнітної сумісності рухомих об'єктів / О.А. Серков, В.В. Князєв, Б.О. Лазуренко, І.В. Яковенко, Г.І. Чурюмов, В.В. Токарєв // Проблеми електромагнітної сумісності перспективних бездротових мереж зв'язку (ЕМС-2019):збірник наукових робіт четвертої міжн. наук.-техн. конф., 24 жовт. 2019 р. - Харків, 2019. - сс. 55-57.

9. Krivoulya G., Tokariev V., Ilina I, Shcherbak V. Mathematical Model for Finding Probability of Detecting Victims of Man-Made Disasters Using Distributed Computer System with Reconfigurable Structure and Programmable Logic / G. Krivoulya, V. Tokariev, I. Ilina, V. Shcherbak // IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology: (PIC S&T), 06-09 oct. 2020y. - Kharkiv, 2020. - pp.197 - 201.

10. Krivoulya G., Tokariev V., Ilina I., Lebediev O., Shcherbak V. Algorithm of Iterations of Distribution of Subtasks Between «S-Bot» in One «Swarm-Bot» System // Proceedings of the 6th International Conference on Computational Linguistics and Intelligent Systems: (COLINS 2022). CEUR Workshop Proceedings., 12-13 may. 2022 y. - Gliwice, Poland. - pp. 1531-1541.

11. Koshevoy N., Ilina I., Tokariev V., Malkova A., Muratov V. Implementation Of The Gravity Search Method For Optimization By Cost Expenses Of Plans For Multifactorial Experiments // Radioelectronic and Computer Systems. – 2023. Vol. 1(105). - pp. 23-32. Doi:

10.32620/reks.2023.1.02.

12. Кривуля Г.Ф., Токарєв В.В., Ільїна І.В., Кравець В.Є. Взаємодія між «s-bots» однієї «Swarm-bot» system у фізичному неорганізованому середовищі, Системи управління, навігації та зв'язку, 2023, №1(71). - сс.108-111. Doi: 10.26906/SUNZ.

13. Krivoulya G., Koshevoy N., Tokariev V., Ilina I., Dubinsky D. Solving the Task of Topological Formation Intelligent Mobile «S-bots» for One «Swarm-bot» System // Proceedings of the 7th International Conference on Computational Linguistics and Intelligent Systems: (COLINS 2023). CEUR Workshop Proceedings., 20-21 april. 2023 y. - Kharkiv, Ukraine. - pp. 273-282.

14. Токарєв В., Ільїна І., Шевченко І., Гриценко І. Про один підхід до рішення асиметричної TSP - задачі при B2C доставках за допомогою платформи "Swarm-bot" - system у фізичному неорганізованому середовищі, Системи управління, навігації та зв'язку, 2023, №4(74), сс. 110-113. Doi: 10.26906/SUNZ.2023.4.110.