Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра математики факультету інформатики

Кваліфікаційна робота
освітня програма - бакалавр
за спеціальністю „Прикладна математика” 113

Планування шляху на основі одометрії

Керівник кваліфікаційної роботи
Курочкін Андрій Володимирович

Старший викладач

_____

(підпис)

Виконала студентка 4-го курсу
спеціальності «Прикладна математика»
Кузьомко Ярина Романівна
«__» _____ 2024 р.

Київ – 2024

Міністерство освіти і науки України

Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»

Кафедра інформатики факультету інформатики

ЗАТВЕРДЖУЮ

Зав.кафедри математики,
проф., д.ф.-м.н.
_____Р. К. Чорней
(підпис)
«___»_____2024 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

на кваліфікаційну роботу

студенту Кузьомко Ярині Романівні факультету інформатики 4 курсу
Тема Планування шляху на основі одометрії
Зміст ТЧ до кваліфікаційної роботи:

Індивідуальне завдання
Перелік прийнятих скорочень
Анотація
Вступ
Розділ 1. Основі засади та види планування переміщення та орієнтації у просторі.
Розділ 2. Аналіз робіт в сфері планування шляху та навігації на певній місцевості за допомогою одометрії.
Розділ 3. Основні підходи та методи роботи в одометрії.
Розділ 4. Реалізація та алгоритми практичної частини.
Висновки
Список літератури

Дата видачі „12” жовтня 2023 р. Керівник _____
(підпис)
Завдання отримав _____
(підпис)

Тема: Планування шляху на основі одометрії
Календарний план виконання роботи:

№ п/п	Назва етапу кваліфікаційної роботи	Термін виконання етапу	Примітка
1.	Отримання завдання на кваліфікаційну роботу.	жовтень 2023	
2.	Обговорення деталей теми на розробка плану виконання	січень-лютий 2024	
3.	Огляд технічної літератури за темою роботи та опрацювання матеріалів	березень 2024	
4.	Написання теоретичної частини роботи	квітень-травень 2024	
5.	Розробка практичної частини та фіналізація практичної	квітень-травень 2024	
6.	Аналіз отриманих результатів, обговорення остаточної доповіді з керівником	травень 2024	
7.	Здача кваліфікаційної роботи	червень 2024	

Студент: Кузьомко Я.Р.

Керівник: Курочкін А.В.

“12” жовтня 2023

ЗМІСТ

ЗМІСТ 4
ПЕРЕЛІК ПРИЙНЯТИХ СКОРОЧЕНЬ 6
АНОТАЦІЯ 7
ВСТУП 8
РОЗДІЛ 1. Основі засади та види планування переміщення та орієнтації у просторі 10
1.1 Інерційна навігаційна система (INS) 11
1.1.1 Інерційний вимірювальний блок (IMU) 11
1.1.2 Робота INS 12
1.2 Орієнтація та навігація за допомогою GPS 12
1.3 LiDAR 14
1.4 Одометрія 15
1.5 Порівняння точності роботи методів орієнтації у просторі 16
РОЗДІЛ 2. Аналіз робіт в сфері планування шляху та навігації на певній місцевості за допомогою одометрії 19
2.1 AutoTrans 19
2.1.1 Математика системи 19
2.1.2 Алгоритм роботи автономного керування квадрокопером 21
2.1.3 Тестування системи 21
2.1.4 Висновки 22
2.2 Reinforcement Learning for the Control of Quadcopters 23
2.2.1 Фізика квадрокоптера 23
2.2.2 Ручне керування 24
2.2.3 PID агент 24
2.2.4 RL агент 25
2.2.5 Результати роботи 26
2.3 On Differential Drive Robot Odometry with Application to Path Planning 27
2.3.1 Одометрія робота 27
2.3.2 Метод калібрування одометрії 28
2.3.3 Заміна кастера на омніколесо 29
2.3.4 Тестування методу 31
2.3.5 Результати роботи 32

РОЗДІЛ 3. Основні підходи та методи роботи в одометрії 33
3.1 Одометрія 33
3.2 Візуальна одометрія 34
РОЗДІЛ 4. Реалізація та алгоритми практичної частини 37
4.1 Постановка задачі 37
4.2 Створення карти та БПЛА 38
4.3 Сенсори руху та їх алгоритм роботи 39
4.3.1 Робота методу sense_obstacles(x, y, angle) 40
4.4 Кінематика руху БПЛА 42
4.5 Метод уникнення перешкод впродовж руху БПЛА avoid_obstacles(point_cloud, no_obstacle_endpoints, target_x, target_y, drone_position) 43
4.5.1 Робота з отриманими даними із сенсорів руху 43
4.5.2 Логіка вибору сторони в яку може рухатись БПЛА 44
4.5.3 Робота методу 47
4.6 Результати виконаної роботи в практичній частині 48
ВИСНОВКИ 53
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 54

ПЕРЕЛІК ПРИЙНЯТИХ СКОРОЧЕНЬ

БПЛА – безпілотний літальний апарат

GPS – система глобального позиціонування

DGPS – диференціальний GPS

RTK-GPS – кінематичний GPS реального часу

LiDAR – light detection and ranging

АНОТАЦІЯ

Кваліфікаційна робота присвячена розробці та вдосконаленню методів одометрії для автономної навігації БПЛА. Основна мета роботи - підвищення точності визначення положення робота за допомогою калібрування одометричних параметрів та впровадження нових підходів до обробки даних сенсорів руху. У роботі детально розглянуто основні види навігаційних систем, включаючи інерційну навігаційну систему, GPS, та методів дистанційного зондування, таких як LiDAR, та їх інтеграцію в алгоритми планування шляху. Практична частина включає створення симуляційної моделі БПЛА та його сенсорної системи для тестування алгоритмів уникнення перешкод та оптимізації траєкторій руху. Результати роботи демонструють підвищення точності позиціонування робота та ефективність запропонованих методів при уникненні перешкод в умовах 2D середовища.

Ключові слова: одометрія, автономна навігація, планування шляху, БПЛА, сенсори руху, уникнення перешкод.

ВСТУП

Сучасний світ активно рухається до автоматизації різноманітних процесів у багатьох сферах життя, таких як виробництво, логістика, медицина, інформаційні технології та навігація. Однією з ключових проблем у розробці автономних систем є забезпечення точного і надійного визначення положення та орієнтації об’єктів у просторі. Одометрія, як один із методів навігації, відіграє важливу роль у цій області завдяки своїй відносній простоті та можливості надання достатньо точних даних для короткотермінових вимірювань.

Зростаюча складність завдань, що виконуються автономними роботами чи літаючими об’єктами, вимагає підвищення точності та надійності роботи алгоритму пересування. Точне визначення положення об’єкта є критично важливим для запобігання зіткнень, оптимізації маршрутів та забезпечення безпеки в умовах невизначеності середовища. Сучасні методи одометрії потребують постійного вдосконалення для зменшення систематичних та випадкових помилок, а також для забезпечення високої точності навіть в умовах значних відстаней та нерівностей поверхні.

Основною метою даної кваліфікаційної роботи є розробка та вдосконалення методів одометрії для підвищення точності визначення положення БПЛА в просторі та оминання перешкод.

Робота складається з чотирьох розділів, кожен з яких висвітлює певні аспекти дослідження. У першому розділі розглядаються основні засади та види планування переміщення та орієнтації у просторі, включаючи огляд основних напрямків автоматизації навігації та переміщення, таких як інерційна навігаційна система, GPS, LiDAR, та методи орієнтації у просторі.

Другий розділ присвячений аналізу наукових робіт, що стосуються планування шляху та навігації мобільних роботів з використанням одометрії.

Тут розглядаються різні підходи і алгоритми, що застосовуються для покращення точності позиціонування.

Третій розділ містить розгляд принципів роботи різних видів одометрії, таких як сенсорна та візуальна одометрія, аналіз їх переваг, недоліків та сфер застосування.

У четвертому розділі описується розробка симуляційної моделі БПЛА, вибір сенсорів та розробка алгоритмів для обробки одометричних даних, а також тестування алгоритмів у створених 2D картах з перешкодами.

У висновках підбиваються підсумки проведених досліджень, оцінюється ефективність запропонованих методів та їх вплив на точність одометрії мобільних роботів. Надаються рекомендації щодо подальших досліджень та вдосконалення розроблених підходів.

РОЗДІЛ 1. Основі засади та види планування переміщення та орієнтації у просторі

За умов швидкого розвитку сучасних технологій все більше і більше сфер та процесів зазнають автоматизації задля покращення продуктивності роботи та зменшенню витрат на виробництво чи обслуговуванню. Основними напрямками є:

- виробництво – роботизовано виробничі лінії, керування та моніторинг процесів, управління інвентарем на складами, автоматизація логістичних процесів;
- інформаційні технології – використання алгоритмів для автоматизації розробки програмного забезпечення, обслуговування вже існуючих проектів, впровадження штучного інтелекту для аналізу даних та їх прогнозування;
- навігація та переміщення – автоматизація руху транспортних засобів, використання різних методів орієнтації в просторі задля автоматизованої навігації, використання систем автоматичного керування рухом для оптимізації потоків транспорту на дорогах, в аеропортах, на морі, тощо.

В наших реаліях автоматизація навігації та переміщенні в просторі є дуже актуальною через низку причин:

- зростання обсягів транспортних засобів, що в свою чергу потребує ефективної навігації та розподілу по дорогам, уникненню заторів та зменшенню часу подорожі
- підвищення безпеки та запобігання аварій за допомогою комп’ютерних алгоритмів та штучного інтелекту
- також не можна забувати про воєнну сферу, де це буде дуже ефективним для наведення на ціль, рух до цілі, а також швидка реакція у рази небезпеки попадання по об’єкту

Орієнтація у просторі є дуже важливою для правильного функціонування, позиціювання та переміщення як БПЛА, так і роботів, і будь-яких транспортних засобів. Залежно від мети, технічних характеристик та можливостей певного об’єкта, для якого потрібно розробити автономну систему навігації, застосовують різні методи орієнтації у просторі. Розглянемо кілька основних способів та в чому полягає їх суть.

1.1 Інерційна навігаційна система (INS)

1.1.1 Інерційний вимірювальний блок (IMU)

Щоб розібратись із алгоритмом роботи INS спершу треба розглянути таке поняття як IMU.

IMU (інерційний вимірювальний блок) – пристрій в певному об’єкті що має певний набір датчиків здатних вимірювати рух[1]. Він містить в собі мінімум три розташовані ортогонально гіроскопи, та три так само розташованих акселерометри. Перші, відповідають за вимірювання швидкості обертання, або ж кутової, навколо кожної осі. Другі – за лінійне прискорення кожної з осей.

Вихідними даними з даного блоку будуть необроблені дані із всіх датчиків, які в подальшому будуть оброблятись вже додатковими пристроями чи системами. Таким чином інформації отриманої з IMU буде достатньо щоб в подальшому визначити орієнтацію об’єкту та напрямок його руху.

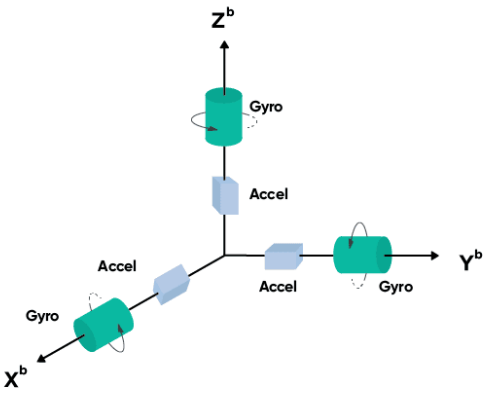

Рис. 1.1.1.1. Схема будови IMU[1]

1.1.2 Робота INS

IMU є вузлом датчиків, який використовується в INS, або інерційній навігаційній системі. INS отримує від IMU вихідні дані, обробляє їх і розраховує зміни у відносному русі об'єкта. Потім INS прив'язує ці зміни до відомої початкової точки, швидкості та напрямку, надаючи положення та векторний вихід в теперішній момент часу t. IMU може бути як інтегрованим у INS напряму, або може бути окремим апаратним засобом, який підключається до зовнішньої INS чи подібної системи. INS обчислює та виводить такі параметри[2]:

- положення, позиційна швидкість і орієнтація у тривимірному просторі;
- лінійна швидкість - векторна величина, що включає в себе як величину, так і напрямок;
- кутова швидкість - швидкість обертання об'єкта навколо своєї осі.

Класичний INS буде включати в себе й інші датчики чи сенсори, щоб отримати вже більш повний набір даних для навігації чи позиціювання[3]:

- GNSS – може одночасно приймати та декодувати сигнали від кількох навігаційних супутників;
- магнітометр – визначає та вимірює силу та напрям магнітних полів;
- датчик тиску – вимірює рівень зовнішнього тиску.

В якості алгоритму обробки даних отриманих INS зі всіх датчиків найчастіше використовується фільтрація Калмана – це алгоритм, що поєднує дані з датчиків і прогнозовані дані[4]. Цей процес відбувається у два етапи, де на першому етапі прогнозується і враховується точність різних вхідних даних, а на другому етапі до вхідних даних застосовується середнє значення.

1.2 Орієнтація та навігація за допомогою GPS

GPS – супутникова навігаційна система, яка дозволяє визначати місцезнаходження об’єкта будь-де на поверхні Землі, чи вище неї[5]. Система

GPS включає групу з 24 активних супутників, що обертаються навколо Землі і передають закодовані радіосигнали. Ці супутники розташовані так, що по чотири з них знаходяться на кожній з шести орбітальних площин, щоб забезпечити постійне покриття усього світу, як показано на рисунку 1.2.1[6].

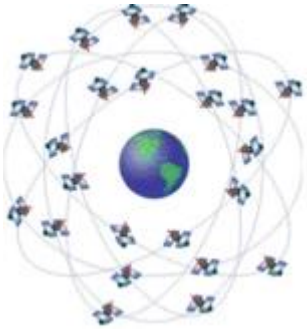

Рис. 1.2.1. Сузір'я супутників GPS навколо Землі[6]

Для визначення місцезнаходження приймача GPS використовується принцип трилатерації, що ґрунтується на вимірюванні відстаней між приймачем і трьома або чотирма супутниками. Цей принцип полягає в тому, що приймач може знаходитись на будь-якому місці на сфері з заданою відстанню від кожного супутника. Перетин цих сфер утворює точку, яка визначає місцезнаходження приймача. Таким чином, приймач може обчислити своє положення, враховуючи відстані до супутників і їхні відомі координати[6].

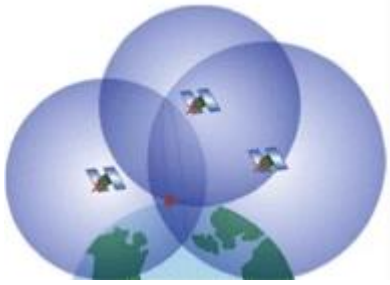

Рис. 1.2.2. Концепція позиціонування за допомогою трилатерації[6]

Стандартний автономний GPS призначений для визначення місцезнаходження з точністю приблизно до 10 метрів. Для підвищення точності GPS і забезпечення локалізації з високою точністю, були розроблені DGPS і RTK-GPS. Ці методи використовують відносне позиціонування, в якому два або

більше приймачів відстежують одні й ті ж супутники одночасно. DGPS складається з трьох основних компонентів: базової станції, розташованої в відомому місці, приймача користувача та каналу зв'язку між ними. DGPS виправляє помилки місцезнаходження приймача користувача, використовуючи виміряні відхилення на базовій станції.

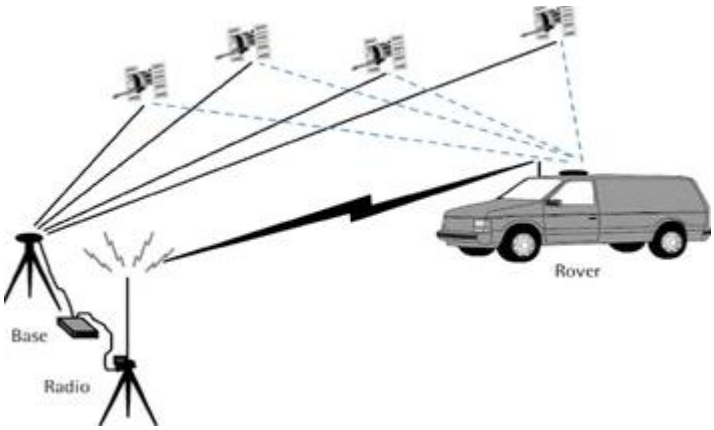

Рис. 1.2.3. Схема роботи DGPS[6]

1.3 LiDAR

LiDAR – це технологія дистанційного зондування для вимірювання відстані, яка передбачає передачу лазера на ціль і подальший аналіз відбитого світла. LiDAR визначає висоту шляхом вимірювання часу, який світло потребує для проходження до поверхні Землі і назад. Цей час використовується для розрахунку відстані, яку світло пройшло. Ця відстань перетворюється на висоту. Лідар використовує ключові компоненти, такі як GPS, для визначення місцезнаходження світлової енергії в просторі, і IMU, щоб визначити орієнтацію літака в повітрі.

Відбита енергія, яка потрапляє назад до датчика, створює форму хвилі, відому як розподіл енергії. Кількість енергії, що повертається до датчика LiDAR, називається «інтенсивністю». Там, де більше фотонів або світлової енергії відбивається до датчика, виникають виразні піки розподілу енергії. Ці піки, що мають форму хвилі, часто відображають об’єкти на поверхні землі, такі як гілки, скупчення листя або будівлі.

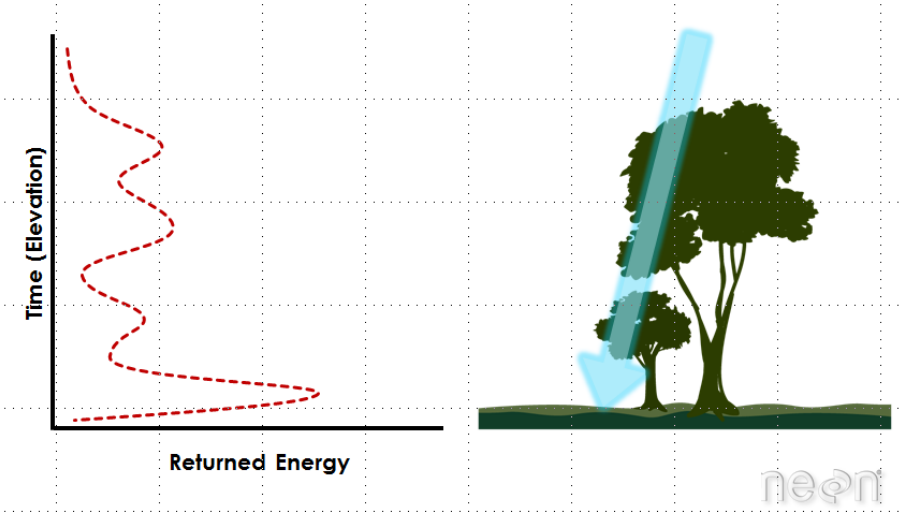

Рис. 1.3.1. Сигнал отриманий із LiDAR[7]

Недоліком LiDAR у порівнянні є висока вартість встановлення. Аналіз даних LiDAR вимагає значних обчислювальних ресурсів, що може вплинути на швидкість реагування реального часу програм. Ітеративний процес обчислення оптимальних відповідностей між лазерними скануваннями збільшує обчислювальну складність. Більше того, сканування може бути неефективним у випадку матеріалів, які пропускають світло, таких як скло, що може спричинити ненадійні дані через відображення на таких поверхнях.

1.4 Одометрія

Одометрія - це використання датчиків руху для визначення того, як робот змінює своє положення відносно початкової точки[8]. Ідея полягає в тому, що позиція змінюється поступово з часом. Зміна положення, виміряна як лінійне зміщення відносно поверхні, може бути визначена на основі обертів колеса. Цей метод надає добру точність для коротких періодів вимірювань, але може призвести до помилок при значних відстанях.

Існує два типи помилок при використанні одометрії:

- систематичні – спричинені недоліками або неточностями в роботі (розвал коліс, неправильно задана їх ширини чи висота, тощо);
- несистематичні – спричинені навколишнім середовищем або помилками в обчисленні (не рівна поверхня, наявність перешкод, не правильно задані рівняння одометрії).

Більш детально принцип роботи цього методу та його види будуть розглянуті у 3 розділі роботи.

1.5 Порівняння точності роботи методів орієнтації у просторі

На основі вище написаного, а також інформації з джерел [1]-[8] можна зробити таблицю-порівняння точності роботи методів:

Метод	Точність	Примітки
GNSS	~10м	Основна точність без додаткових корекцій
DGPS	до 1м	Використання двох або більше приймачів для зменшення похибок
RTK-GPS	до 2см	Вимагає мінімум п'ять супутників для максимальної точності
Одометрія	Варіантивна (залежить від поверхні)	Схильна до накопичення похибок через ковзання коліс
Візуальна одометрія	Достатньо висока	Точність залежить від якості зображень та обчислювальних потужностей
LiDAR	Висока	Висока точність у визначенні відстаней, але висока вартість та обчислювальні витрати
Інтегровані системи	дуже висока	Поєднання різних технологій для покращення точності та надійності в складних

		умовах

Табл. 1.5.1. Порівняння різних методів визначення орієнтації

Аналізуючи точність різних навігаційних систем, можна побачити значні відмінності в їх можливостях та областях застосування. Стандартний GNSS, який забезпечує точність близько 10 метрів, є базовим рішенням для навігації, але цього буде не достатньо для випадків чи систем, де потрібна висока точність. Для покращення цієї точності використовуються вдосконалені версії, такі як диференціальний GPS (DGPS) та RTK-GPS. DGPS може досягати точності до 1 метра завдяки використанню двох або більше приймачів, що дозволяє зменшити похибки. RTK-GPS забезпечує точність до 2 сантиметрів, але для цього необхідно щонайменше п'ять супутників, що робить його ідеальним для додатків, що потребують високої точності, таких як сільське господарство або будівництво.

Одометрія, що базується на вимірюванні обертів коліс, може мати варіативну точність, яка залежить від умов поверхні та може накопичувати похибки на довгих відстанях. Це робить її менш надійною в умовах, де необхідна висока точність. Проте, ця технологія є простою та дешевою, що робить її привабливою для деяких застосувань.

Візуальна одометрія використовує камери для аналізу зображень та визначення змін положення. Вона може забезпечувати високу точність, особливо в умовах, де GNSS-сигнали недоступні або ненадійні. Однак, точність VO залежить від якості зображень та обчислювальних потужностей, що може вимагати значних ресурсів.

LIDAR забезпечує дуже високу точність вимірювання відстаней (до сантиметрів) та використовується для створення детальних карт і тривимірного моделювання. Проте, ця технологія є дорогою та потребує значних обчислювальних ресурсів для аналізу даних, що може обмежувати її використання в реальному часі.

Інтегровані системи, що поєднують GNSS з візуальною одометрією або LIDAR, можуть досягати дуже високої точності за рахунок використання переваг кожної технології та компенсації їх недоліків. Такі системи забезпечують найбільшу точність та надійність у складних умовах, роблячи їх ідеальними для вимогливих завдань, таких як автономні транспортні засоби або роботизовані системи.

Таким чином, вибір навігаційної системи залежить від конкретних вимог до точності та умов використання. Для завдань, що потребують високої точності, краще використовувати RTK-GPS або інтегровані системи. Для менш вимогливих додатків може бути достатньо стандартного GNSS або одометрії, в той час як візуальна одометрія та LIDAR підійдуть для специфічних середовищ, де інші системи можуть бути менш ефективними.

РОЗДІЛ 2. Аналіз робіт в сфері планування шляху та навігації на певній місцевості за допомогою одометрії

Одометрія широко застосовується в плануванні шляху та орієнтації в просторі чи на певній карті. В цьому розділі буде розглянуто три дуже показові та великі роботи із використанням цього методу визначення положення.

2.1 AutoTrans

Стаття[9] присвячена розробці та оцінці комплексної системи планування та контролю для БПЛА з підвішеними вантажами. Основною метою є створення методів, що забезпечують автономне транспортування вантажів БПЛА, враховуючи динамічні особливості та зовнішні перешкоди на шляху.

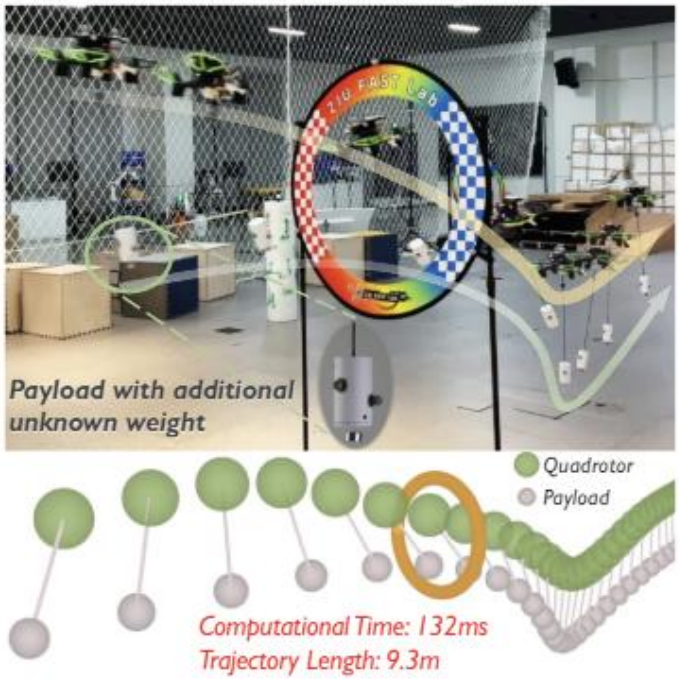

Рис. 2.1.1. Квадрокоптер, оснащений підвісною системою вантажу, пролітає через вузький прохід. Комп'ютер квадрокоптера генерує траєкторію проліту за 132 мс, при цьому довжина прохідного шляху становить 9.3 м.[9]

2.1.1 Математика системи

Для забезпечення роботи алгоритму китайські вчені використовували декілька основних математичних формул. Динаміку всієї системи, що враховує взаємодію квадрокоптера із підвісним вантажем описує[9]:

$$\left(m_Q + m_L\right)(\ddot{x}_L + ge_z^w) = \left(\rho \cdot fRe_z^w - m_Q l(\dot{\rho} \cdot \dot{\rho})\right)\rho$$

$$m_Q l\ddot{\rho} + m_Q l(\dot{\rho} \cdot \dot{\rho})\rho = \rho \times (\rho \times fRe_z^w)$$

$$x_Q = x_L - l\rho$$

$$\dot{R} = R\breve{w}, \tau = J\dot{\omega} + \omega \times J\omega$$

де,

- x_Q, x_L – центри маси квадрокоптера та підвісної ваги відповідно;
- $\ddot{x}_L$ – прискорення підвісної ваги;
- ρ – кут нахилу кабеля на якому висить вага;
- l – довжина кабеля;
- J – матриця енерції квадрокоптера;
- ω – кутова швидкість квадрокоптера;
- f – сила тяжіння квадрокоптера;
- R – кут обертання квадрокоптера;
- τ – ветор моментів квадрокоптера;
- e_z^w – вісь Z світової системи координат.

Диференціальна площина системи задається формулами[9]:

$$m_L(\ddot{x}_L + ge_z^w) = -f_c\rho, \qquad \rho = -\frac{(\ddot{x}_L + ge_z^w)}{\|(\ddot{x}_L + ge_z^w)\|_2^2}$$

$$fRe_z^w = m_Q\left(\ddot{x}_Q + ge_z^w\right) - f_c\rho$$

А також не менш важлива формула що визначає задачу оптимізації для адаптивного NMPC. Цільова функція мінімізує відхилення від заданої траєкторії та керуючих сигналів, враховуючи обмеження системи[9]:

$$\min_u J = \frac{1}{2}\sum_{k+1}^{N}\left(\|\tilde{x}(k)\|_{Q_k}^2 + \|\tilde{u}(k)\|_{H_k}^2\right) + \frac{1}{2}\|\tilde{x}(N+1)\|_{Q_e}^2$$

$$x(k+1) = f\left(x(k), u(k)\right)$$

$$u_{min} \le u \le u_{max}, x(N+1) \in X_f$$

2.1.2 Алгоритм роботи автономного керування квадрокопером

Алгоритм автономного керування БПЛА включає планування траєкторії в реальному часі та адаптивну систему управління (NMPC). Планування генерує плавні траєкторії, враховуючи змінну форму та нелінійні характеристики системи. Для цього використовується диференціальна площина, що перетворює обмеження на безпеку та динамічну здійсненність у диференційовані функції штрафів. Задача оптимізації вирішується просторово-часовим оптимізатором траєкторій, а початкові рішення знаходяться динамічним пошуком.

Система управління (NMPC) з ієрархічною компенсацією збитків забезпечує точне відстеження траєкторій, враховуючи зовнішні перешкоди та неточності моделі. Вона містить оцінювач зовнішніх сил та інкрементну нелінійну динамічну інверсію (INDI) для контролера кутової швидкості.

Процес роботи алгоритму починається з моделювання системи як жорсткого тіла з натягнутим кабелем. Диференціальна плоскість зменшує розмірність задачі оптимізації, визначаючи стан системи через плоскі виходи, такі як позиція вантажу та кут нахилу квадрокптера. Спеціально-часова оптимізація формулює задачі уникнення зіткнень і динамічної здійсненності, використовуючи модель сферичних бульбашок для змінної форми системи.

Алгоритм забезпечує безпеку та високу точність відстеження траєкторій навіть за наявності зовнішніх перешкод та невизначеності.

2.1.3 Тестування системи

Тестування системи автономного керування квадрокоптером проводилося через серію симуляційних та реальних експериментів для оцінки ефективності планування траєкторій і точності управління в умовах зовнішніх перешкод та невизначеності.

У симуляціях перевіряли планування траєкторій у трьох різних сценаріях: проста 3D середа з 12 випадковими квадратами, проходження через випадковий отвір і складне середовище із великою кількістю перешкод. Було створено 10 симуляційних середовищ для кожного сценарію, і результати показали

найвищий рівень успішності та швидкості обчислень порівняно з іншими методами.

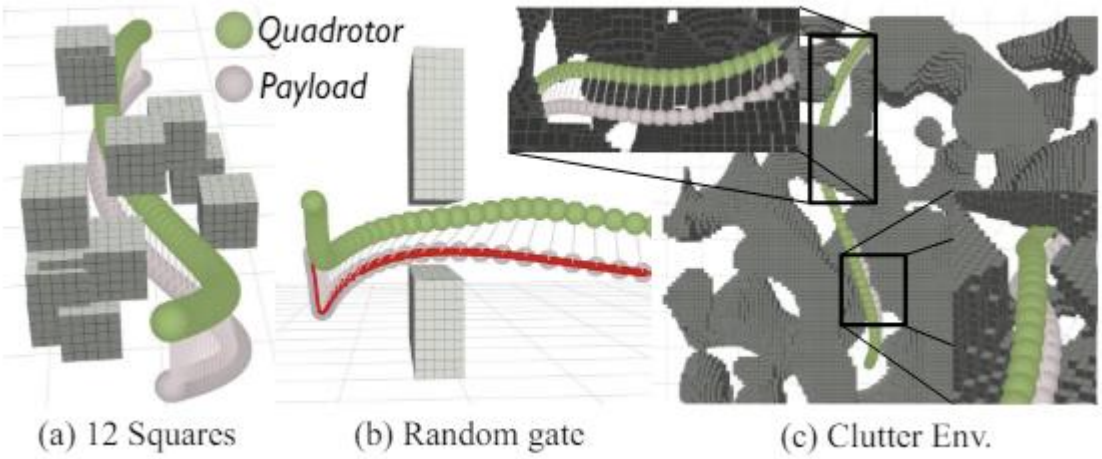

Рис. 2.1.3.1. Середовища для тестування алгоритму[9]

У реальних умовах контролер тестували на восьми траєкторіях з різними профілями швидкості та прискорення. Для оцінки впливу невідомих параметрів на вантаж підвішували додаткові ваги (50г та 200г). Контролер також перевіряли на вітростійкість за допомогою вентиляторів, що створювали вітрові потоки зі швидкістю 4.5 м/с та 9 м/с. У всіх випадках система управління показала високу точність.

В реальних експериментах квадрокоптер з підвішеним вантажем успішно проходив через ворота на різних швидкостях та з різними маневрами, демонструючи здатність генерувати динамічно здійсненні траєкторії в реальному часі. У складних та невідомих середовищах квадрокоптер, оснащений камерою для сприйняття оточення, швидко адаптувався до нових перешкод, що вимагало постійного перепланування траєкторій для забезпечення безпеки.

2.1.4 Висновки

Було розроблено ефективний метод генерації траєкторії, який відповідає вимогам безпеки і працює в режимі реального часу, а також контролер NMPC з ієрархічною стратегією компенсації перешкод для протидії впливу зовнішніх

факторів. Було проведено велику кількість експериментів, які підтверджують ефективність і переваги нашого підходу.

2.2 Reinforcement Learning for the Control of Quadcopters

У цій роботі[10] автор порівнює три методи управління квадрокоптером для навігації та досягнення цілі без аварій:

- ручне керування – квадрокоптером керує людина використовуючи клавіатуру;
- PID агент – квадрокоптер автономно орієнтується відносно своїх маршрутних точок, використовуючи позицію квадрокоптера та позицію маршрутних точок як вхідні дані;
- RL(reinforcement learning) агент – у цьому методі квадрокоптер самостійно рухається до маршрутних точок, генеруючи силу тяжіння роторів на основі свого положення і положення точок маршруту через політику агента, який навчався в подібних умовах.

Розглянемо детальніше запропоновану фізику квадрокоптера, у чому полягав кожен з методів та як автор це реалізовував, а також проаналізуємо отримані результати.

2.2.1 Фізика квадрокоптера

Крок у момент часу t для обчислення положення квадрокоптера в момент часу $t+1$ є наступними[10]:

$$\ddot{x} = \frac{-(T_l + T_r)\sin(\theta)}{m}$$

$$\ddot{y} = \frac{-(T_l + T_r)\cos(\theta)}{m} + g$$

$$\ddot{\theta} = \frac{(T_r - T_l) \times l}{m}$$

де,

- $\ddot{x}$, $\ddot{y}$, $\ddot{\theta}$ – прискорення квадрокоптера (по вісі X, по вісі У, та куту нахилу θ відповідно);

- T_l, T_r – значення сили тяжіння для лівого та правого ротора відповідно;
- θ – кут нахилу квадрокоптера по вісі Z.
- m, g, l – константи що відповідають масі, гравітації та розміру квадрокоптера.

2.2.2 Ручне керування

Тут все дуже просто – гравець керує квадрокоптером за допомогою клавіш вгору, вниз, вправо, вліво на клавіатурі, що в свою чергу змінює значення T_l, T_r що в свою чергу змінює напрямок, кут нахилу та швидкість квадрокоптера[10].

- початкові значення сил тяжіння для роторів – $T_l = 0.04$, $T_r = 0.04$;
- робота клавіші «вгору» – $T_l += 0.04$, $T_r += 0.04$;
- робота клавіші «вниз» – $T_l -= 0.04$, $T_r -= 0.04$;
- робота клавіші «вправо» – $T_l += 0.003$, $T_r -= 0.003$;
- робота клавіші «вліво» – $T_l -= 0.003$, $T_r += 0.003$;

2.2.3 PID агент

Управління тут базується на позиційній різниці між квадрокоптером і цільовою точкою. Вертикальна похибка використовується для регулювання тяги обох роторів. Наприклад, якщо обидва ротори працюють на 0,08 потужності, квадрокоптер піднімається вгору.

Вертикальна відстань між квадрокоптером і цільовою точкою передається на PID-контролер для розрахунку оптимальної вертикальної швидкості.

Різниця між поточною та оптимальною вертикальними швидкостями передається на PID-контролер для регулювання амплітуди тяги, яка рівномірно подається на обидва ротори (від 0,08 до 0).

Похибка по горизонталі використовується для контролю різниці у тязі кожного ротора. Наприклад, якщо лівий ротор працює на -0.003, а правий на +0.003, квадрокоптер нахиляється вліво і рухається в цю сторону.

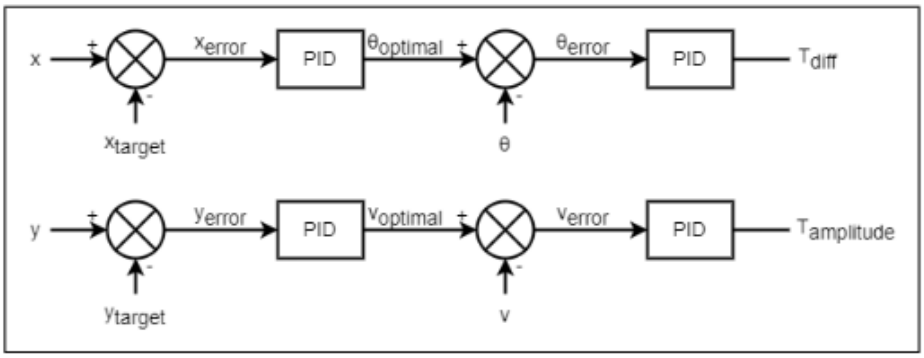

Рис. 2.2.3.1. Схема роботи PID агента[10]

2.2.4 RL агент

Для успішного навчання та прийняття рішень цього агента потрібна наступна інформація із квадрокоптера в кожен момент часу t:

- відстань до цілі;
- кут нахилу до цілі;
- кут між нахилом квадрокоптера до вісі У;
- швидкість квадрокоптера;
- кут між вектором швидкості і віссю У;
- кут між вектором швидкості та вектором від квадрокоптера до цілі.

Далі навчання агента відбувається наступним чином – за кожен момент часу t залежно від дії агента йому буде нараховуватись певна винагорода або покарання, якщо виконана дія була не правильною[10]:

- $+\frac{1}{60}$ – нагорода за виживання (квадрокоптер не вилетів за межі карти);
- $-\frac{dist}{100\times60}$ – покарання за перебування далеко від цілі ($dist$ – відстань до цілі);
- $+100$ – нагорода за досягення цілі;
- -1000 – покарання за те що квадрокоптер вилетів за межі карти.

Обране просторове поле для дій виглядає наступним чином: агент може вибирати між двома плаваючими значеннями, які позначаються як action0 та action1. Перший параметр (action0) визначає амплітуду тяжіння, що

застосовується до обох роторів, а другий параметр (action1) визначає відносну різницю у тяжінні, що застосовується до обох роторів.

Значення тяжіння можна розрахувати з вибраних дій за допомогою таких формул[10]:

- $T_l = action0 - 0{,}04 + action1 - 0{,}0006 + 0{,}04$
- $T_r = action0 - 0{,}04 - action1 - 0{,}0006 + 0{,}04$

Навчання проводилося протягом 3,3 мільйонів кроків в першому поколінні, та 5 мільйонів в другому.

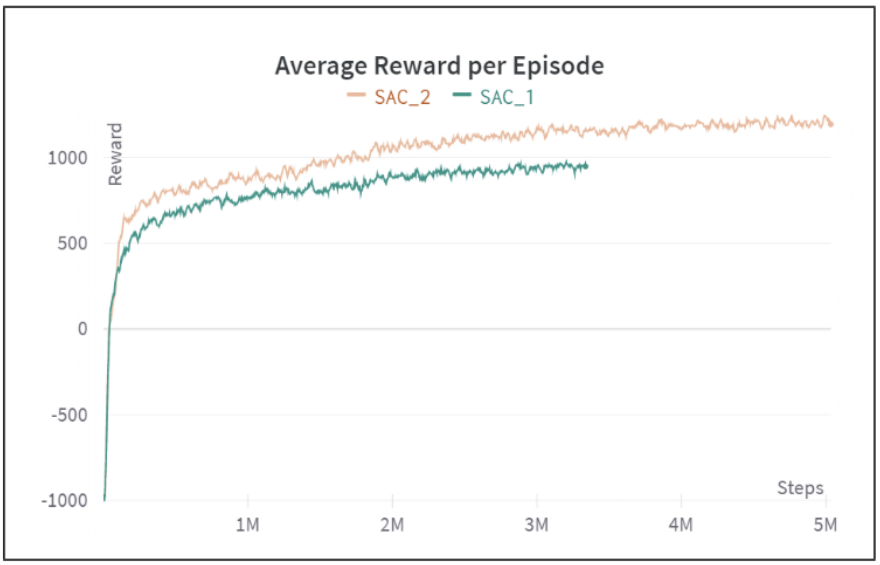

Рис. 2.2.4.1. Результати навчання двох поколінь RL агента[10]

2.2.5 Результати роботи

В результаті автор запустив одночасно всі 3 методи та дав 100 секунд на збір якомога більшої кількості цілей, кількість набраних балів показано на таблиці 2.2.5.1:

Агент	Бали
Людина	51
PID	55
SAC	65

Табл. 2.2.5.1. Результати роботи методів[10]

Людина схильна до помилок і може пропустити ціль та не бути готовою до наступного кроку.

PID відзначається великою стабільністю, він дуже консервативно переслідує цілі, що в свою чергу гарантує відсутність зіткнень та помилок, проте час потрібний на роботу сильно збільшується.

SAC не робить помилок, він адаптується до будь-якої ситуації і реагує на нові цілі дуже швидко та точно, що робить його в результаті найкращим агентом.

2.3 On Differential Drive Robot Odometry with Application to Path Planning

У цій роботі розглядається покращення орієнтації мобільного робота в момент часу t за допомогою методу калібрування одометрії та заміни кастера на омніколесо[11]. Описується вплив систематичних помилок одометрії та пропонуються методи їх мінімізації. Проведено експерименти для оцінки точності одометрії та ефективності запропонованих методів. Тепер проаналізуємо виконану роботу більше детально.

2.3.1 Одометрія робота

Робот Pioneer 3 DX оснащений двома керованими колесами з енкодерами та опорним колесом для стабільності. Основні параметри одометрії - це радіуси правого та лівого коліс R_r, R_l відповідно та відстань між колесами – D. Лінійна швидкість робота V_s визначається як[11]:

$$V_s = \frac{\omega_r R_r + \omega_l R_l}{2}$$

$$\omega = \frac{\omega_r R_r - \omega_l R_l}{D}$$

де, ω_r, ω_l – кутові швидкості правого та лівого коліс відповідно.

Тоді кінематичні рівняння одометрії мають наступний вигляд[11]:

$$\dot{x} = V_s \cos(\theta)$$

$$\dot{y} = V_s \sin(\theta)$$

$$\dot{\theta} = \omega$$

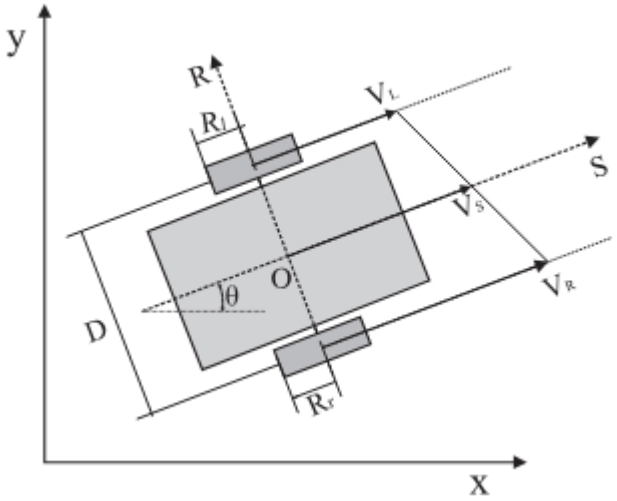

Рис. 2.3.1.1. Схематичне зображення робота[11]

2.3.2 Метод калібрування одометрії

Даний метод базується на інтеграції кінематичних рівнянь для точного визначення основних параметрів одометрії: радіусів коліс R_r, R_l та відстані між колесами D. У ньому враховуються систематичні і випадкові похибки, щоб покращити точність оцінок параметрів.

Позначаємо $\Delta R_r, \Delta R_l, \Delta D$ як оцінки відповідних параметрів, тоді їх похибки будуть мати вигляд[11]:

$$\Delta R_r = \widehat{R_r} - R_r$$

$$\Delta R_l = \widehat{R_l} - R_l$$

$$\Delta D = \widehat{D} - D$$

Похибки в методі виникають через неточності у вимірюванні радіусів коліс та відстані між ними. Вони враховуються у вигляді помилок у лінійних та кутових швидкостях, які розраховуються за формулами[11]:

$$x' = (V_S \cos\theta)\left(\frac{\Delta R_r}{R_r} + \frac{\Delta R_l}{R_l} + \frac{\Delta D}{D}\right)$$

$$y' = (V_S \sin\theta)\left(\frac{\Delta R_r}{R_r} + \frac{\Delta R_l}{R_l} + \frac{\Delta D}{D}\right)$$

Систематичні похибки також виникають через неточності у моделі робота, такі як некаліброваність коліс та неточне визначення початкового кута робота. Для моделювання цих похибок вводиться додатковий параметр α, який представляє початковий кут між поздовжньою віссю робота та віссю глобальної системи координат[11]:

$$\dot{x} = V_s \cos(\theta + \alpha)$$
$$\dot{y} = V_s \sin(\theta + \alpha)$$

2.3.3 Заміна кастера на омніколесо

В ході роботи було виявлено, що систематичні помилки одометрії виникають через використання кастера, оскільки вісь обертання кастера не є абсолютно вертикальною як показано на рисунку 2.3.3.1., як це передбачено.

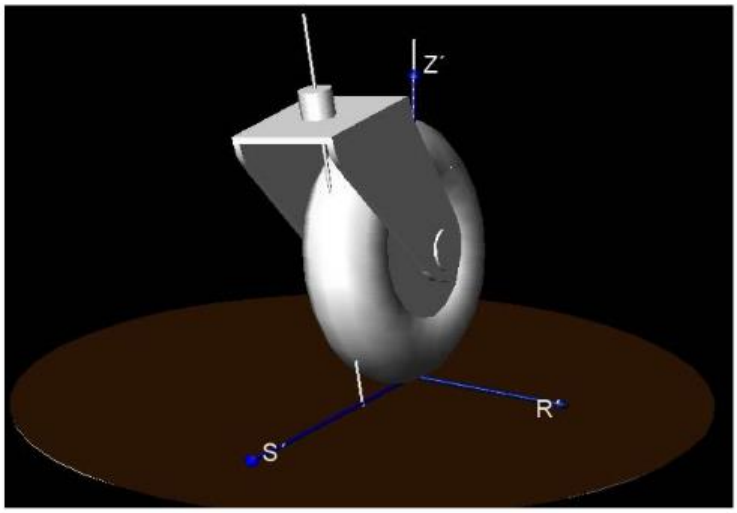

Рис. 2.3.3.1. Вісь обертання кастера[11]

Реакція F ґрунту на колесо розкладається на два компоненти – F_1 та F_2. F_2 є паралельною до осі кастера, тоді як F_1 перпендикулярна до неї і намагається обертати кастер. Як тільки кастер повертається, з'являється сила тертя T, що видно на рисунку 2.3.3.2., яка намагається зробити кастер паралельним до площини руху. Хоча ця сила мала, вона впливає на робота протягом усього шляху.

Оскільки моделювання цього впливу є складним завданням з невизначеним практичним результатом, було прийняте рішення замінити

кастер на омніколесо, тобто колесо, яке може рухатися в напрямках, паралельних до його осі обертання, без ковзання що продемонстровано на рисунку 2.3.3.3.

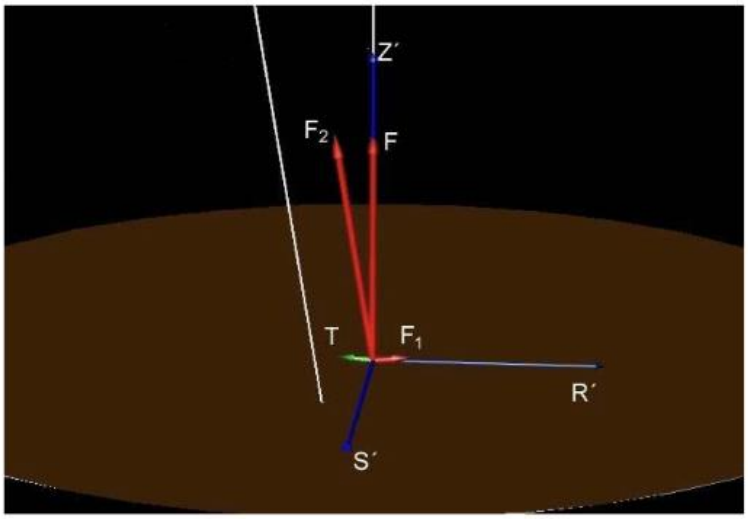

Рис. 2.3.3.2. Поява сили тертя[11]

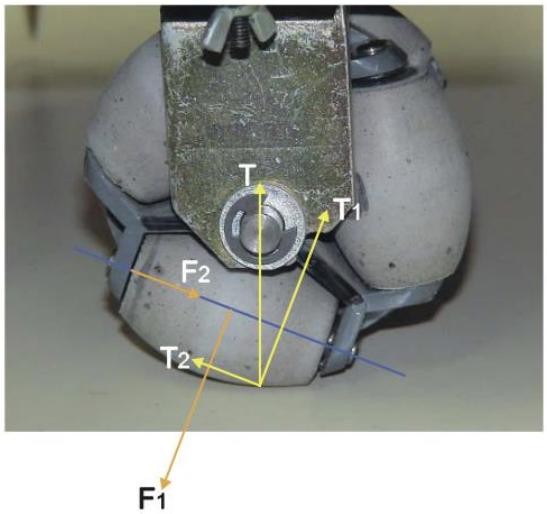

Рис. 2.3.3.3. Схема омніколеса[11]

Основна перевага омніколеса полягає в тому, що воно діє як точка опори з незначним зсувом і що описані вище систематичні похибки мінімізуються. Ще одна суттєва перевага полягає в тому, що збурювальні сили, викликані омніколесом, є випадковими, і, отже, результуючі помилки є не систематичними і тому менш серйозними для методу одометрії.

2.3.4 Тестування методу

Чотири групи вимірювань були проведені на мобільному роботі Pioneer 3 DX з наступними цілями:

- калібрування одометрії;
- визначення значень R_r, R_l, D;
- оцінка покращення одометрії після калібрування та заміни кастерного колеса на омніколесо.

Кожна група включала п’ять різних траєкторій для розрахунку параметрів R_r, R_l, D, а також одну кінцеву траєкторію для оцінки точності отриманих даних. У всіх випадках робот стартував з однакового початкового положення та рухався по рівній поверхні. Після кожної траєкторії фіксували кінцеве положення та орієнтацію робота, і визначалися зміщення x, y і θ.

Калібрувальні траєкторії були довільної форми і довжини. Після проходження всіх п'яти траєкторій, за допомогою методу калібрування одометрії, обчислювалися помилки$\Delta R_r, \Delta R_l, \Delta D$ і θ, а також їхні скориговані значення.

У перших двох групах експериментів використовували кастерне колесо, тоді як у третій і четвертій групах його замінили на омніколесо. Крім того, у першій і третій групах траєкторії мали радіуси кривизни більше одного метра, а у другій і четвертій – менше одного метра, включаючи повороти на місці.

Номер групи	Помилка одометрії до калібрування(см)	Помилка одометрії після калібрування(см)	Середня довжина шляху(м)	Примітки
1	35.59	4.17	126.5	$r > 1m$, кастер
2	60.87	44.48	124.9	$r < 1m$, кастер
3	16.10	7.44	126.2	$r > 1m$, омніколесо
4	37.03	14.14	124.8	$r > 1m$, омніколесо

Табл. 2.3.4.1. Результати експериментів[11]

2.3.5 Результати роботи

У цій статті були розроблені нові методи для роботів з диференціальним приводом. Спочатку було досліджено точність орієнтації роботів у просторі, що використовують одометрію. Для вирішення проблеми орієнтації був застосований метод калібрування одометрії. Помилки одометрії інтегрувалися вздовж всього шляху, і були розраховані нові, покращені параметри одометрії. Метод передбачав використання різних початкових і кінцевих точок без застосування додаткових датчиків, окрім енкодерів. Важливою перевагою цього методу є його універсальність для будь-якої моделі одометрії. Дослідження також показало, що заміна кастера на омніколесо забезпечує кращу точність одометрії.

РОЗДІЛ 3. Основні підходи та методи роботи в одометрії

Із загальним використанням одометрії та підходами реалізації проектів на цій базі було проаналізовано в попердньому розділі, тепер більш детально розглянемо види одометрії та принцип їх роботи на прикладах.

3.1 Одометрія

Першим методом є одометрія, або ж сенсорна одометрія – це спосіб навігації та позиціонування у просторі за допомогою даних із сенсорів руху[12]. Саме слово утворене від грецьких слів одос (маршрут) і метрон (вимірювання), що і є найкращим описом суті методики. Зачасту такий спосіб використовують саме для роботів, де одометр вимірює кількість обертів коліс або інших рухомих частин транспортного засобу і обчислює відстань, пройдену від початкової точки до точки в теперішній момент.

Розглянемо принцип такої роботи на найпростішому роботі з двома колесами. Для нього мають місце позначення лінійної відстані D_r та D_l, що позначають відстань, яку пройшло кожне колесо, також W – ширина робота, що вимірюється як відстань між двома точками контакту робота до поверхні, початкова орієнтація (кут нахилу) робота задана в радіанах – $O_{(t)}$. На рисунку 3.1.1 зображена схема руху робота за одиницю часу t.

Нова орієнтація робота $O_{(t+1)}$ буде обчислюватись за формулою[12]:

$$O_{(t+1)} = O_{(t)} + \frac{(D_r + D_l)}{W}$$

Відстань, пройдена за час $t + 1$[12]:

$$D_{t,t+1} = \frac{(D_r + D_l)}{2}$$

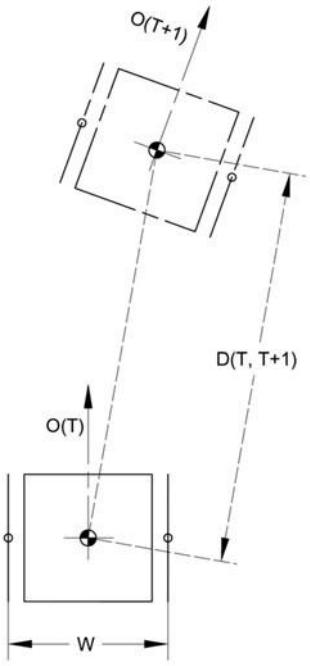

Рис. 3.1.1. Схематичний рух робота за одиницю часу t[12]

Такий метод буде дуже дієвим за умови переміщення на коротких відстанях в стабільних умовах із швидким збором потрібної інформації із сенсорів руху. Однак на не рівних поверхнях, чи недостатнього дотику коліс до поверхні можуть виникнути проблеми із точністю вимірювань, що призведе до неправильної роботи робота. Також цей метод є чутливим до помилок, через постійно інтеграцію швидкості впродовж часу руху.

3.2 Візуальна одометрія

Другим дуже поширеним методом є візуальна одометрія – визначення положення та подальша навігація за допомогою зображень, отриманих із камер на транспортному засобі. Такий підхід активно використовують NASA в своїх марсоходах.

Найчастіше для визначення зміни положення на основі візуальних даних використовують метод відповідності шаблону[13]-[14]. Його суть полягає в тому, зіставляють два послідовних зображення отримані з камери та визначається вертикальне (Δv) та горизонтальне(Δu) зміщення пікселів між ними. Після цього за допомогою формул (1)-(2) знайдене зміщення пікселів

перетворюють на фізичне горизонтальне чи вертикальне зміщення камери на транспортному об’єкті[15]:

$$\Delta X_c = -\Delta u \left(\frac{Z_c}{f_c}\right) \quad (1)$$

$$\Delta Y_c = -\Delta v \left(\frac{Z_c}{f_y}\right) \quad (2)$$

де, $(X_c,\ Y_c,\ Z_c)$ – система координат камери.

Щоб трансформувати систему координат камери $(X_c,\ Y_c,\ Z_c)$ в систему координат транспортного засобу $(X_v,\ Y_v,\ Z_v)$ розраховують матрицю обертання R_c – обертають камеру на 180° навколо осі Z, а потім на 180° навколо осі Y (3)-(4)[15]:

$$R_c = R_z \times R_y = \begin{bmatrix} \cos(\theta_z) & -\sin(\theta_z) & 0 \\ \sin(\theta_z) & \cos(\theta_z) & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} \cos(\theta_y) & 0 & \sin(\theta_y) \\ 0 & 1 & 0 \\ -\sin(\theta_y) & 0 & \cos(\theta_y) \end{bmatrix} \quad (3)$$

$$\begin{bmatrix} \Delta X_{v_i} \\ \Delta Y_{v_i} \\ \Delta Z_{v_i} \end{bmatrix} = R_c \times \begin{bmatrix} \Delta X_{c_i} \\ \Delta Y_{c_i} \\ \Delta Z_{c_i} \end{bmatrix} \quad (4)$$

де, θ – кут обертання камери.

Оскільки передбачається, що модель руху є керованою моделлю Акермана, із трансляцією ΔX та кутом обертання $\Delta\theta$, то в координатній площині транспортного засобу це буде обчислюватись за допомогою рівняння[15]:

$$\Delta X_i = \Delta X_{v_i}$$

$$\Delta\theta_i = tan^{-1}(\frac{\Delta Y_i}{L_{cam}})$$

де, L_{cam} – відстань між центром обертання транспортного засобу та центром обертання камери, ΔX_v та ΔY_v – переміщення транспортного засобу в своїй системі координат.

В результаті нове положення транспортного засобу P_{new} буде обчислювати за формулою[15]:

$$P_{new} = P_{prev} + R_z \times T_{inc}$$

де, P_{prev} – попереднє положення, R_z – матриця повороту навколо осі Z, T_{inc} – поступове переміщення по осі X.

Схематичне пояснення алгоритму роботи методу відповідності шаблонів наведено на рисунку 3.2.1.

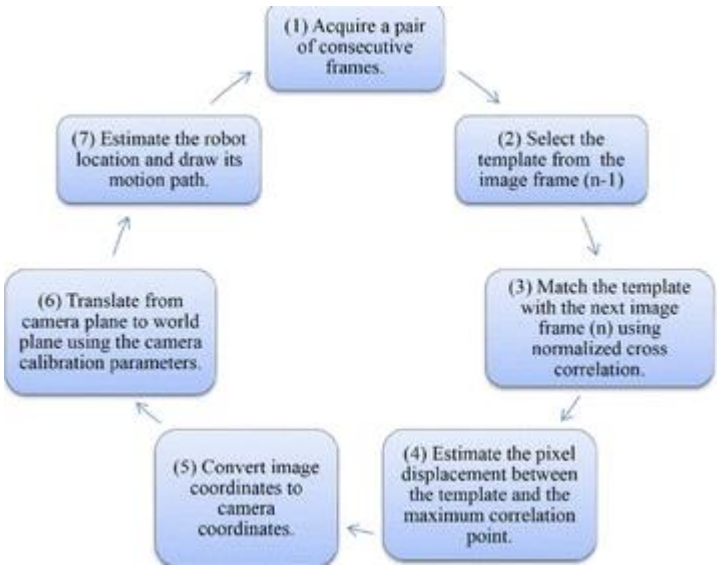

Рис. 3.2.1. Алгоритм роботи методу відповідності шаблонів[15]

РОЗДІЛ 4. Реалізація та алгоритми практичної частини

4.1 Постановка задачі

У даній роботі розглядається задача автономного польоту БПЛА у середовищі з перешкодами. БПЛА оснащений сенсорами руху, які дозволяють йому отримувати інформацію про оточення, зокрема про наявність та розташування перешкод. Для навігації БПЛА використовує методи одометрії, які дозволяють йому орієнтуватись у просторі та переміщатись.

Основна задача практичної частини полягає у розробці алгоритму, який забезпечить такі умови:

- виявлення та обминання перешкод на шляху до цілі;
- постійне коригування траєкторії польоту у реальному часі на основі даних сенсорів руху;
- досягнення заданої цілі без врізання чи вильотів за межі карти.

Розробка практичної частини включає наступні кроки:

- аналіз літератури та схожих робіт, що було описано в 2 розділі роботи;
- вибір сенсорів – вибір та створення відповідних сенсорів, що будуть збирати та зберігати всю необхідну інформацію впродовж кожного моменту часу t;
- математичне моделювання руху БПЛА з урахуванням даних із сенсорів руху;
- розробка алгоритму, що буде забезпечувати пересування БПЛА по карті уникаючи існуючі перешкоди;
- тестування алгоритму на заданих 2D картах із перешкодами та цілями, а також відслідковування пройденого БПЛА маршруту.

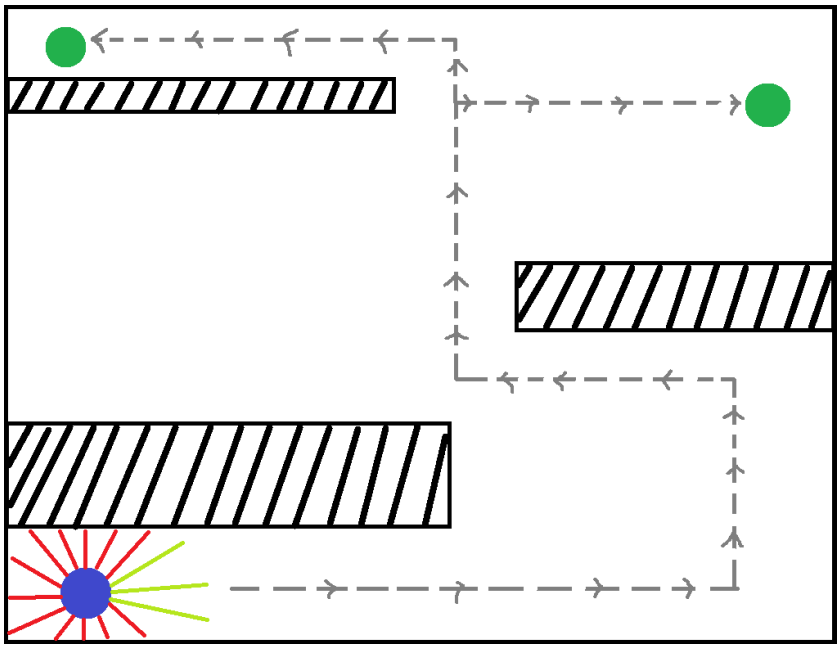

Рис. 4.1.1. Схематична постановка задачі для практичної реалізації

На рис. 4.1.1. схематично зображено постановку задачі – БПЛА в нижньому правому кутку карти, має певні сенсори руху що визначають чи є поруч перешкоди, далі орієнтуючись по даним із цих сенсорів БПЛА має визначати напрямок куди він може полетіти в даних момент часу t. В результаті БПЛА має долетіти до цілі, що позначені зеленим, керуючись розробленим алгоритмом.

Очікується, що розроблений алгоритм забезпечить високий рівень автономності польоту БПЛА, дозволяючи йому ефективно та безпечно досягати визначеної цілі, оминаючи перешкоди. Результати дослідження можуть бути використані для подальшого вдосконалення систем автономного управління БПЛА, а також для подальшого створення алгоритмів польоту БПЛА вже у 3D просторі.

4.2 Створення карти та БПЛА

Для цього завдання було створено дві карти в 2D просторі із перешкодами для тестування алгоритму польоту БПЛА. Перша має розміри 800×1000 пікселів, друга - 1300×900 пікселів. Карти мають перешкоди у вигляді чорних блоків, що обмежують рух БПЛА з різних сторін.

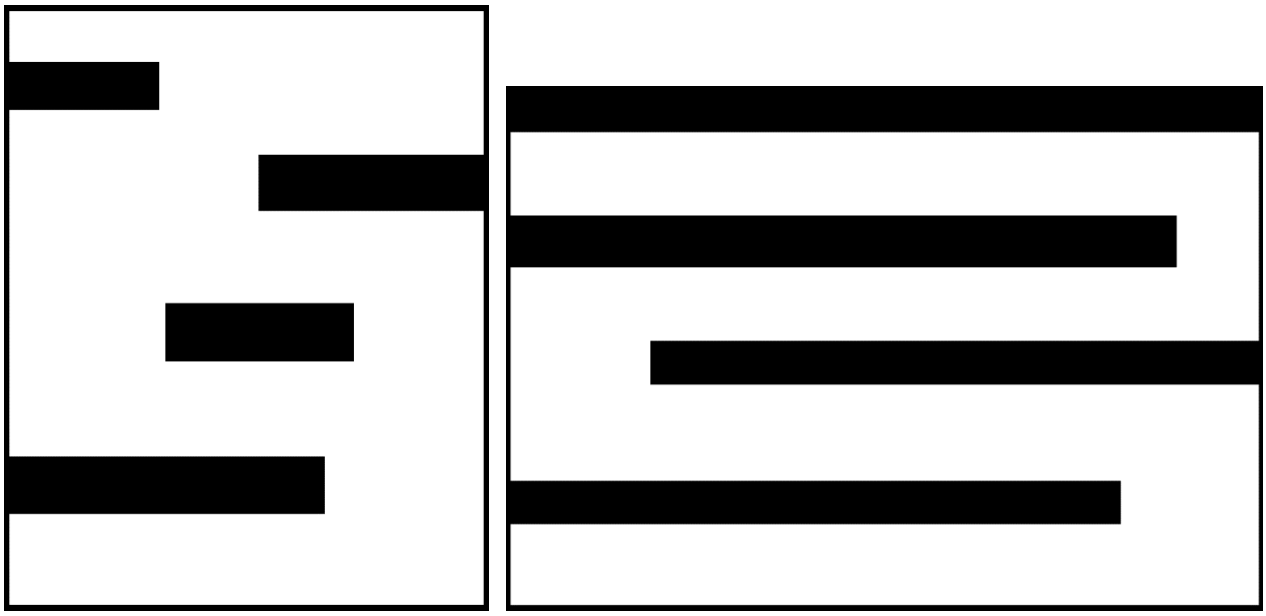

Рис. 4.2.1. Перша та друга карти відповідно[16]

Власне БПЛА має розміри 68×27 пікселів. БПЛА має 2 простих пропелера, які мають значення сили тяжіння T_l, T_r – для лівого та правого пропелерів відповідно. Кінематика БПЛА буде базуватись на цих двох значеннях, та за допомогою рівнянь одометрії визначатись як і куди БПЛА полетить.

рис. 4.2.2. Зображення БПЛА [16]

4.3 Сенсори руху та їх алгоритм роботи

Однієї із найбільш важливої частин практичної роботи було створення сенсорів руху, що будуть збирати всю необхідну інформацію про розташування перешкод. У випадку цієї роботи сенсорами є імітовані лазерні сенсори, що розташовані по колу навколо БПЛА.

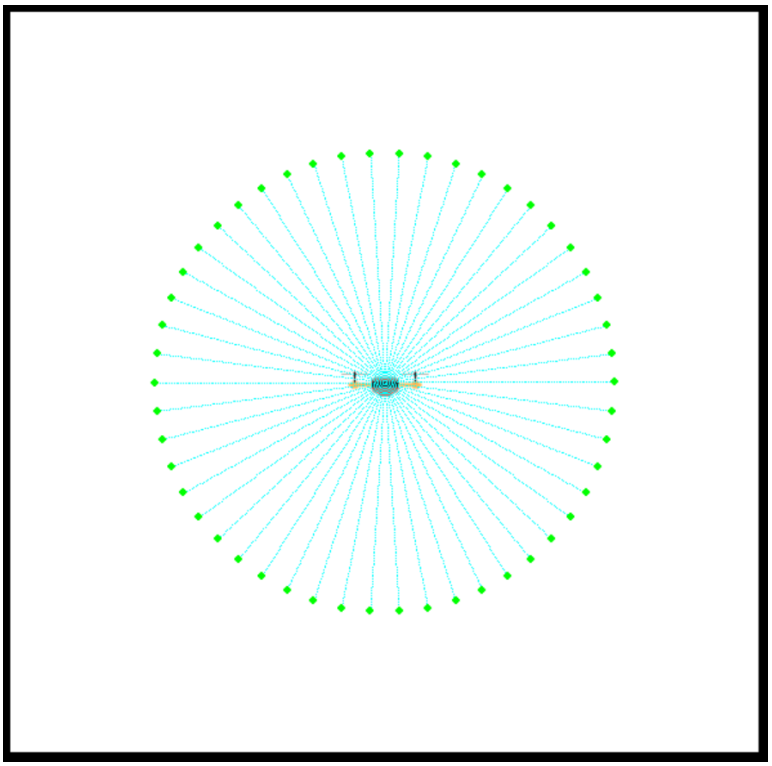

Рис. 4.3.1. Сенсори руху БПЛА

Всього БПЛА має 50 сенсорів розташованих по колу, що у свою чергу мають максимальну довжину в 200 пікселів. Упродовж роботи алгоритму сенсори розділяють на дві групи:

- obstacles_endpoints – сенсори, що дотикаються до перешкоди;
- no_obstacle_endpoints – сенсори, що не знаходять перешкоду по всій свої довжині.

4.3.1 Робота методу sense_obstacles(x, y, angle)

Метод $sense_obstacles$[16] призначений для виявлення перешкод навколо БПЛА на основі його поточного положення – координати x та y, та кута повороту $angle$. Він повертає два списки: $obstacles_endpoints$, який містить координати та кути виявлених перешкод, та $no_obstacle_endpoints$, який містить координати та кути сенсорів без перешкод.

Спочатку метод визначає початкову та кінцеву величини кута сканування – $start_angle$ і $finish_angle$, які враховують поточний кут повороту БПЛА та радіус дії сенсора $sensor_range$. Далі, використовуючи ці кути, метод

проходить по кожному з них з кроком, визначеним кількістю сенсорів – $SENSOR_COUNT$.

Для кожного кута обчислюються кінцеві координати $x2, y2$ променя сенсора. Потім, у циклі, метод поступово переміщується від початкових координат БПЛА $x1, y1$ до кінцевих координат променя з кроком $u = \frac{i}{100}$, де $i = range(0, SENSOR_LEN)$, який визначає позиції між початком і кінцем променя.

Якщо координати потрапляють в межі мапи, метод перевіряє колір пікселя в цих координатах на карті. Якщо колір пікселя чорний, це означає, що перешкода виявлена, і відповідні координати та кут додаються до списку $obstacles_endpoints$. Якщо перешкода не виявлена на жодному з відрізків променя, координати та кут додаються до списку $no_obstacle_endpoints$.

Таким чином, метод сканує навколишню область робота та виявляє перешкоди, додаючи їхні координати та кути до відповідних списків, які потім повертаються для подальшого використання.

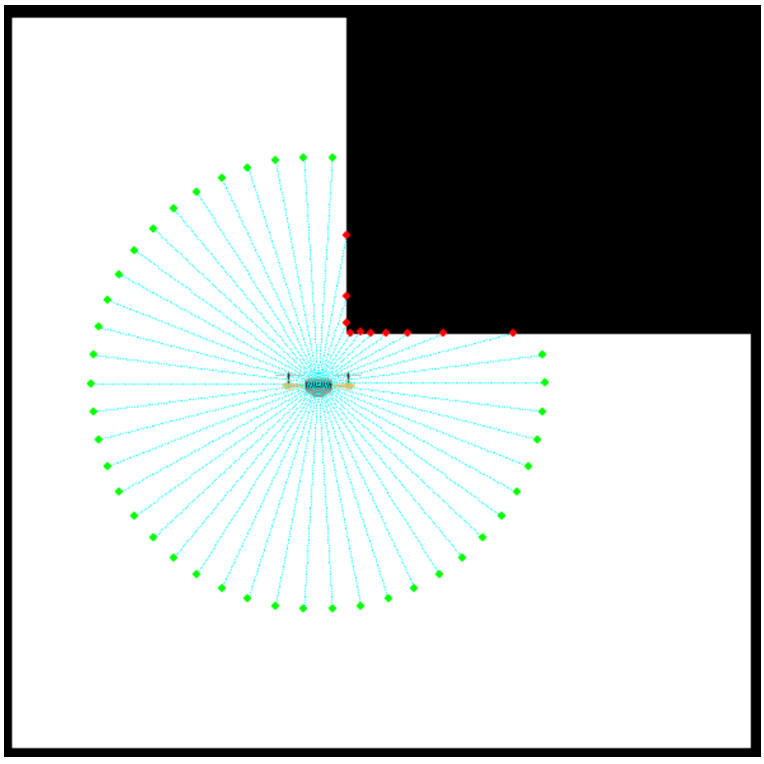

Рис. 4.3.1.1. Робота методу sense_obstacles(x, y, angle)

На рисунку 4.3.1.1. видно як саме метод визначає перешкоди, а також підкреслює які саме сенсори бачать біля себе перешкоду. Якщо вивести логування методу, то отримаємо наступні значення:

- $obstacles_endpoints = 10$;
- $no_obstacle_endpoints = 40$.

Порахувавши промені на рисунку 4.3.1.1. побачимо, що метод дійсно правильно визначає наявність та кількість перешкод.

4.4 Кінематика руху БПЛА

Рівняння одометрії для руху БПЛА в 3D просторі задаються наступним чином:

$$\ddot{x} = \frac{-(T_l + T_r)\sin(\theta)}{m}$$

$$\ddot{y} = \frac{-(T_l + T_r)\cos(\theta)}{m} + g$$

$$\ddot{\theta} = \frac{(T_r - T_l) \times l}{m}$$

де,

- $\ddot{x}$, $\ddot{y}$, $\ddot{\theta}$ – прискорення БПЛА (по вісі X, по вісі У, та куту нахилу θ відповідно);
- T_l, T_r – значення сили тяжіння для лівого та правого ротора відповідно;
- θ – кут нахилу БПЛА по вісі Z.
- m, g, l – константи що відповідають масі, гравітації та розміру БПЛА.

Проте, оскільки робота виконана для 2D простору, то ми можемо знехтувати масою, гравітацією та кутом нахилу по вісі Z. Тоді, формули набувають вигляду:

$$\ddot{x} = \frac{-(T_l + T_r)}{2}$$

$$\ddot{y} = \frac{-(T_l + T_r)}{2}$$

4.5 Метод уникнення перешкод впродовж руху БПЛА avoid_obstacles(point_cloud, no_obstacle_endpoints, target_x, target_y, drone_position)

Цей метод[16] є найбільшим та найвагомішим для роботи всієї програми. За його допомогою та результатів методу $sense_obstacles(x, y, angle)$ БПЛА визначає куди йому летіти в залежності від наявності чи відсутності перешкод в певній стороні. Розглянемо детальніше як він працює.

4.5.1 Робота з отриманими даними із сенсорів руху

Спершу метод опрацьовує дані отримані із сенсорів руху, а саме $point_cloud$ – масив із координатами на кутами перешкод, $no_obstacle_endpoints$ – масив координат та кутів сенсорів, що не виявили перешкоди.

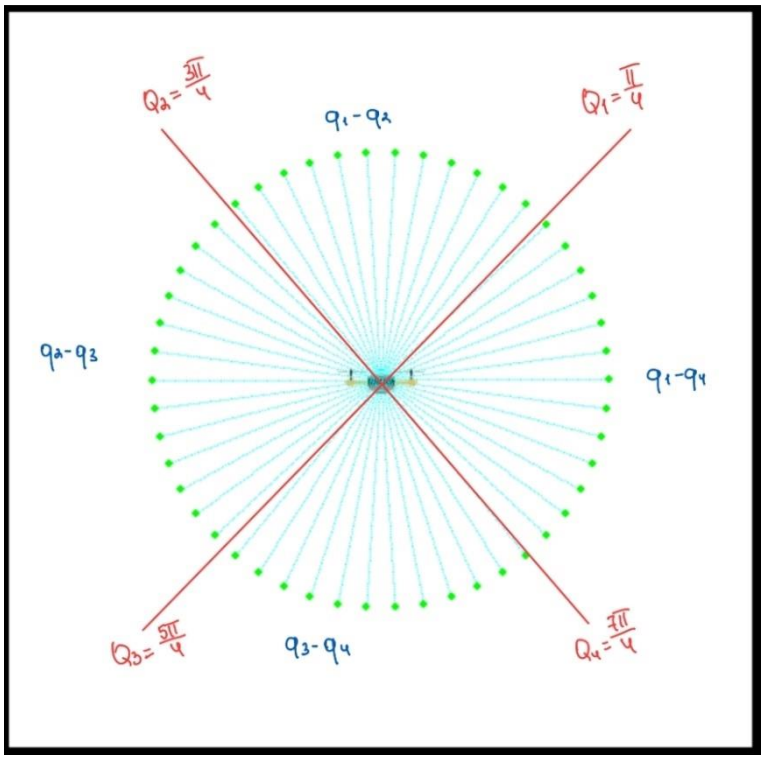

Рис. 4.5.1.1. Розподіл на квадранти

Спочатку створюється словник $counts_obs$, який підраховує кількість перешкод у кожному з чотирьох квадрантів як показано на рисунку 4.5.1.1.

Квадранти визначаються кутами $Q_1 = \frac{\pi}{4}, Q_2 = \frac{3\pi}{4}, Q_3 = \frac{5\pi}{4}$ та $Q_4 = \frac{7\pi}{4}$. Кількість перешкод підраховується для кожного квадранта окремо:

- квадрант $q1_q2$ містить перешкоди між кутами Q_1 та Q_2, відповідає за верх БПЛА;
- квадрант $q2_q3$ — між кутами Q_2 та Q_3, відповідає за ліву сторону БПЛА;
- квадрант $q3_q4$ — між кутами Q_3 та Q_4, відповідає за низ БПЛА;
- квадрант $q4_q1$ містить перешкоди, які знаходяться або менше ніж Q_1, або більше ніж Q_4, відповідає за праву сторону БПЛА.

Після цього визначається квадрант з найменшою кількістю перешкод, і виводиться його назва – min_key, та кількість перешкод – min_value.

Далі створюється аналогічний словник $counts_no_obs$, який підраховує кількість сенсорів що не виявили перешкоди у кожному квадранті. Процедура підрахунку аналогічна до тієї, що використовується для сенсорів, що виявили перешкоди. Потім визначається квадрант з найбільшою кількістю безперешкодних точок, і виводиться його назва – max_key, та кількість безперешкодних точок – max_value.

Якщо у хмарі точок $point_cloud$ більше однієї перешкоди, починається перевірка відстаней до кожної з них. Для кожної точки в $point_cloud$ обчислюється відстань від поточної позиції БПЛА до цієї точки та знаходиться найменша відстань, що зберігається у змінну $closest_obs$, яка містить координати точки кінця променя, його кут та власне значення відстані.

4.5.2 Логіка вибору сторони в яку може рухатись БПЛА

Найважливішим було правильно обчислити значення кута для сенсорів, що визначають чи може БПЛА піднятись вверх або спуститись вниз. Для цього до ширини БПЛА було додано по 5 пікселів з кожної сторони, задля забезпечення безпечної відстані підняття чи спуску біля перешкоди і це значення дорівнює дузі кола, що має обмежуватись двома сенсорами, кути нахилу яких і будуть

шуканими значеннями. Далі методи $angles_up$ та $angles_down$ визначають значення цих кутів за допомогою наступних формул для верхніх кутів:

$$start_ang = \frac{\pi}{2} - \left\lfloor \frac{\lfloor \alpha / \beta \rfloor}{2} \right\rfloor \cdot \alpha$$

$$end_ang = \frac{\pi}{2} + \left\lfloor \frac{\lfloor \alpha / \beta \rfloor}{2} \right\rfloor \cdot \alpha$$

А для нижніх мають вигляд:

$$start_ang = \frac{3\pi}{2} - \left\lfloor \frac{\lfloor \alpha / \beta \rfloor}{2} \right\rfloor \cdot \alpha$$

$$end_ang = \frac{3\pi}{2} + \left\lfloor \frac{\lfloor \alpha / \beta \rfloor}{2} \right\rfloor \cdot \alpha$$

де, $\alpha = \frac{2\pi}{SENSOR_COUNT}$, $\beta = \frac{arc_len}{SENSOR_LEN}$.

У даних формулах α відповідає куту між двома сусідніми сенсорами, а β – центральний кут, що відповідає дузі кола необхідній для польоту БПЛА вверх чи вниз.

Отже, в методі $angles_up$ спочатку обчислюється довжина дуги, додаючи 10 одиниць до ширини БПЛА. Потім визначається центральний кут для цієї дуги, враховуючи довжину дуги arc_len та довжину сенсора що дорівнює радіусу кола $SENSOR_LEN$. Далі визначається значення кута між двома сусідніми сенсорами за допомогою загальної кількості сенсорів $SENSOR_COUNT$. Потім визначається, на який кут припадає половина цієї дуги, І вже останній описаний кут використовується для обчислення початкового та кінцевого кутів, які окреслюють дугу: початковий кут отримується шляхом віднімання половини кута дуги від $\frac{\pi}{2}$, а кінцевий кут — шляхом додавання

половини кута дуги до $\frac{\pi}{2}$. Метод $angles_down$ працює аналогічно, лише в кінці використовується значення $\frac{3\pi}{2}$ замість $\frac{\pi}{2}$.

Вибір таких значень кута в радіанах, що використовується у формулах, зумовлено тим, що середнє арифметичне між ними має кутом, який є строго перпендикулярний до осі X. Це якраз і дозволить точно визначити чи вистачить місця для оминання перешкоди в процесі руху вгору чи вниз.

В результаті обох методів ми отримуємо, що кількість сенсорів, що не бачать перед собою перешкод згори чи знизу, дорівнює шести. Далі використовуються методи can_go_up та can_go_down, що в свою чергу обчислюють скільки точок із квадрантів $q1_q2$ та $q3_q4$, для верху та низу відповідно відповідають визначеним кутам у методах $angles_up$ та $angles_down$.

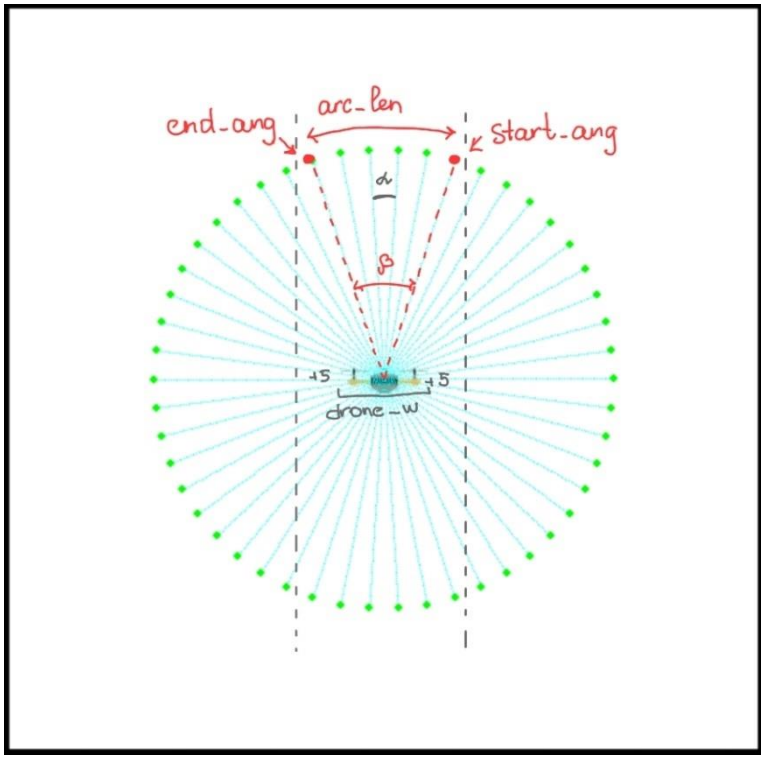

Рис. 4.5.2.1. Схематичне позначення кутів $start_ang$ та end_ang на прикладі методу $angles_up$

Визначення траєкторію польоту вправо чи вліво відбувається наступним чином:

- якщо $counts_obs['q4_q1'] < counts_obs['q2_q3']$, то БПЛА летить вправо;
- якщо навпаки – вліво.

Тут логіка простіше – за допомогою описаного в підрозділі 4.5.1 аналізу даних отриманих із сенсорів, визначається в якому із квадрантів менше точок перешкод, той напрямок і обирається.

4.5.3 Робота методу

На цьому етапі вводиться змінна $is_direction_selected$, що відповідає за визначення напрямку руху БПЛА вздовж осі X, тобто вліво або вправо.

Наступний крок – перевірка знаходження цілі відносно БПЛА. За умови, що координата $y_{target} < y_{drone}$, то задається мета летіти вверх при першій можливості, і навпаки – летіти вниз, якщо $y_{target} > y_{drone}$.

Розглянемо алгоритм роботи методу[16] на прикладі коли БПЛА задана мета летіти вверх, тобто ціль знаходиться вище його самого.

Спочатку перевіряється, чи знаходиться поточна позиція БПЛА по осі Y в межах ±10 одиниць від цільової координати по осі Y. Якщо це так, то БПЛА перевіряє координати по осі X. Якщо і по осі X він знаходиться в межах ±10 одиниць від цільової координати – БПЛА досяг цілі. Якщо ж ні, то БПЛА починає рухатися вліво або вправо, залежно від того, де знаходиться цільова точка по осі X.

Якщо БПЛА не знаходиться в межах ±10 одиниць від цільової координати по осі Y, він перевіряє відстань до найближчої перешкоди. Якщо ця відстань більша за мінімальну дозволену відстань min_obs_dist, і перешкода знаходиться в першому квадранті, БПЛА рухається вгору. Якщо перешкода знаходиться в другому або четвертому квадранті, БПЛА також рухається вгору.

Якщо ж напрямок ще не вибрано, БПЛА обирає його залежно від кількості перешкод в кожному квадранті як описано у попередньому підрозділі.

Якщо БПЛА вже вибрав напрямок, він перевіряє, чи може він рухатися вгору, використовуючи метод can_go_up. Якщо кількість точок повернутих методом $can_go_up = 6$ – БПЛА рухається вгору. Якщо ні, БПЛА починає рух в обраний напрямок: якщо напрямок праворуч, БПЛА перевіряє, чи може він продовжувати рух праворуч, і якщо ні, змінює напрямок на ліворуч. Аналогічно, якщо напрямок ліворуч, БПЛА перевіряє можливість руху ліворуч, і якщо це небезпечно, змінює напрямок на праворуч.

Таким чином, БПЛА постійно аналізує навколишнє середовище, перевіряючи відстані до перешкод та напрямки руху, щоб вибрати найбезпечніший шлях до цільової точки.

Коли БПЛА задана ціль летіти вниз, тобто ціль знаходить нижче за нього, то робота методу відбувається аналогічно, за винятком того що замість методу can_go_up, буде перевірятись can_go_down та БПЛА буде летіти вниз.

4.6 Результати виконаної роботи в практичній частині

Тестування написаного алгоритму відбувалось на двох картах в 2D просторі, що були описані в підрозділі 4.2.

Для кожної карти перевірялись наступні шість варіантів:

- маршрут №1: початкова позиція БПЛА – лівий нижній кут карти, ціль розташована в правому верхньому куті;
- маршрут №2: початкова позиція БПЛА – лівий нижній кут карти, ціль розташована в лівому верхньому куті;
- маршрут №3: початкова позиція БПЛА – правий верхній кут карти, ціль розташована в лівому нижньому куті;
- маршрут №4: початкова позиція БПЛА – лівий верхній кут карти, ціль розташована в правому нижньому куті;
- маршрут №5: початкова позиція БПЛА – центр карти, ціль розташована в лівому верхньому куті;

- маршрут №6: початкова позиція БПЛА – центр карти, ціль розташована в правому нижньому куті.

Результати роботи програми на першій карті зображені на рисунках 4.6.1.-4.6.6:

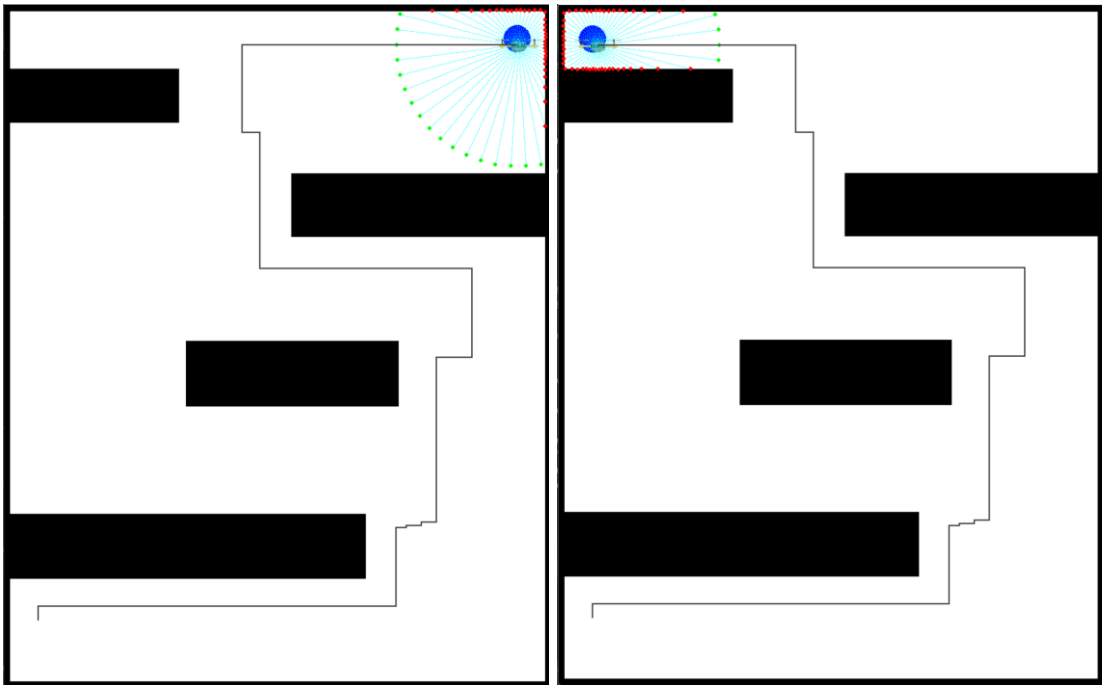

Рис. 4.6.1. та 4.6.2. Результати проходження маршрутів №1 та №2 відповідно

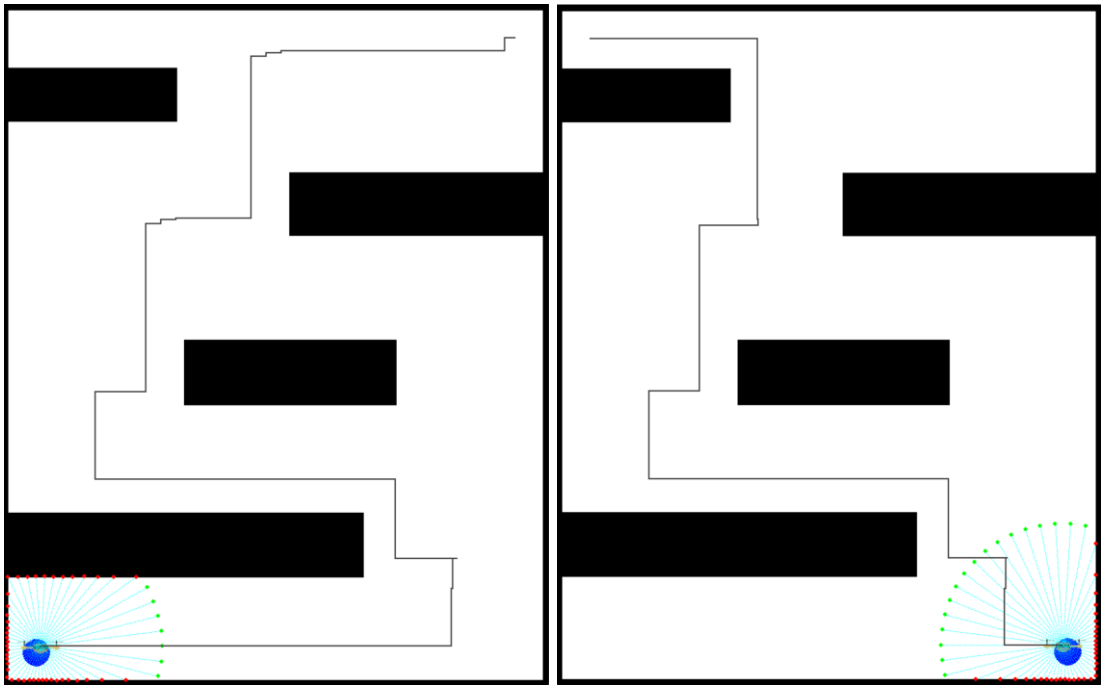

Рис. 4.6.3. та 4.6.4. Результати проходження маршрутів №3 та №4 відповідно

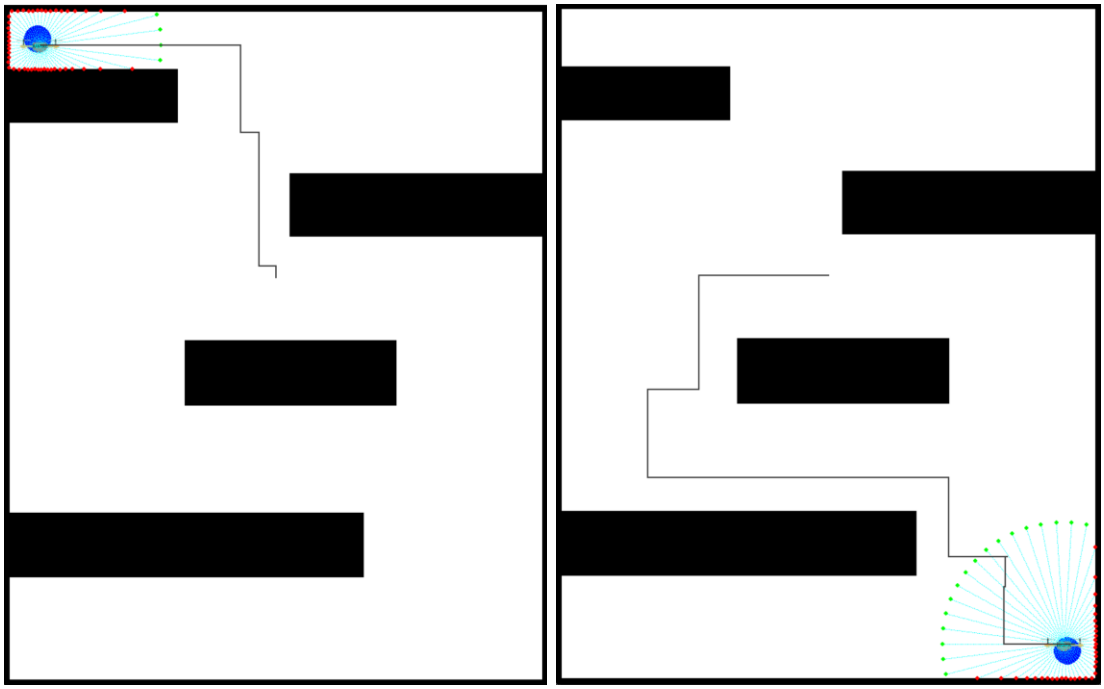

Рис. 4.6.5. та 4.6.6. Результати проходження маршрутів №5 та №6 відповідно

Результати роботи програми на другій карті зображені на рисунках 4.6.7.-4.6.12:

Рис. 4.6.7. Результат проходження маршруту №1

Рис. 4.6.8. Результат проходження маршруту №2

Рис. 4.6.9. Результат проходження маршруту №3

Рис. 4.6.10. Результат проходження маршруту №4

Рис. 4.6.11. Результат проходження маршруту №5

Рис. 4.6.12. Результат проходження маршруту №6

Отже, як бачимо, основні задачі що виносились на практичну частину роботи виконані – БПЛА автономно пересувається в просторі, визначає перешкоди, коригує траєкторію польоту впродовж всього руху та досягає цілі не врізаючись нікуди та не вилітаючи за межі карти.

ВИСНОВКИ

Підсумовуючи все вищенаписане, можна зробити висновок, що мета роботи була виконана. В результаті проведених досліджень та реалізації алгоритмів ця було створено алгоритм підвищення точності визначення положення БПЛА у 2D просторі за допомогою вдосконалення методів одометрії та сенсорів руху.

Теоретична частина роботи включала розгляд основних засад та видів планування переміщення та орієнтації у просторі, огляд сучасних навігаційних систем, таких як інерційна навігаційна система, GPS та LiDAR. Проведено аналіз наукових робіт у сфері планування шляху та навігації, а також розглянуто різні підходи та методи роботи в одометрії, такі як сенсорна та візуальна одометрія.

Практична частина роботи включала створення симуляційної моделі БПЛА та його сенсорної системи для тестування алгоритмів уникнення перешкод та оптимізації траєкторій руху. Було розроблено та реалізовано алгоритм, який забезпечує виявлення та обминання перешкод, постійне коригування траєкторії польоту у реальному часі та досягнення заданої цілі без врізання чи вильотів за межі карти. Для тестування було створено дві 2D карти з перешкодами, а також симуляційну модель БПЛА з сенсорами руху. Розроблені алгоритми виявлення перешкод, кінематики руху БПЛА та уникнення перешкод показали високу ефективність в умовах змодельованого середовища. Алгоритми були протестовані на двох картах з шістьма різними маршрутами. Результати показали, що БПЛА успішно досягає цілі, уникаючи перешкод та коригуючи траєкторію у реальному часі.

Отже, результати проведеної роботи підтверджують ефективність запропонованих методів і алгоритмів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Inertial Measurement Unit (IMU) [Електронний ресурс] // Advanced Navigation – Режим доступу до ресурсу: https://www.advancednavigation.com/tech-articles/inertial-measurement-unit-imu-an-introduction/.
2. Smith G. What Is an Inertial Navigation System? [Електронний ресурс] / Grant Maloy Smith. – 2023. – Режим доступу до ресурсу: https://dewesoft.com/blog/what-is-inertial-navigation-system.
3. Inertial Navigation Systems (INS) [Електронний ресурс] // Advanced Navigation – Режим доступу до ресурсу: https://www.advancednavigation.com/tech-articles/inertial-navigation-systems-ins-an-introduction/.
4. Fauzi, Hilman, Batool, Uzma "A Three-bar Truss Design using Single-solution Simulated Kalman Filter Optimizer". Mekatronika. 1 (2)., - 15 July 2019., - с. 98–102.
5. Cook G. Mobile Robots: Navigation, Control and Remote Sensing / Gerald Cook. – Нью Йорк, 2011. – (Institute of Electrical and Electronics Engineers).
6. Georgy, A. N. T. B. K. (2013). Jacques. Fundamentals of Inertial navigation, Satellite-based Positioning and their Integration.
7. Wasser L. The Basics of LiDAR - Light Detection and Ranging - Remote Sensing / Leah A. Wasser., 2024.
8. https://www.geeksforgeeks.org/odometry/ [Електронний ресурс]. – 2021. – Режим доступу до ресурсу: https://www.geeksforgeeks.org/odometry/.
9. AutoTrans: A Complete Planning and Control Framework for Autonomous UAV Payload Transportation [Електронний ресурс] / H.Li, H. Wang, C. Feng2, C. Gao. – 2023. – Режим доступу до ресурсу: https://arxiv.org/pdf/2310.15050.
10. Sajus A. Reinforcement Learning for the Control of Quadcopters [Електронний ресурс] / Alexandre Sajus – Режим доступу до ресурсу:

https://github.com/AlexandreSajus/Quadcopter-AI/blob/main/Reinforcement_Learning_for_the_Control_of_Quadcopters.pdf.

11. Papadopoulos E. On Differential Drive Robot Odometry with Application to Path Planning [Електронний ресурс] / E. Papadopoulos, M. Misailidis // Proceedings of the European Control Conference. – 2007. – Режим доступу до ресурсу: https://www.researchgate.net/publication/242389452_On_Differential_Drive_Robot_Odometry_with_Application_to_Path_Planning.
12. Odometry [Електронний ресурс] // SimReal. – 2007. – Режим доступу до ресурсу: https://web.archive.org/web/20150215184240/http://simreal.com/content/Odometry
13. Choi M. A novel two stage template matching method for rotation and illumination invariance / M. Choi, W. Kim // Pattern Recognition / M. Choi, W. Kim., 2002. – С. 119–129.
14. Goshtasby A, Gage SH, Bartholic JF / Template Matching Techniques in Computer Vision: Theory and Practice. IEEE Trans Pattern Anal Mach Intell., 1984. – С. 374–378.
15. Review of visual odometry: types, approaches, challenges, and applications / Mohammad O. A. Aqel, Mohammad H. Marhaban, M. Iqbal Saripan, Napsiah Bt. Ismail. // SpringerPlus. – 2016. – №1897.
16. Kuzomko Y. Odometry based path planning [Електронний ресурс] / Yaryna Kuzomko – Режим доступу до ресурсу: https://github.com/YarynaKuziomko/diploma_code.