

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ
ТЕХНОЛОГІЙ**

**НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ**

КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Система автономного керування для малогабаритного
транспортного засобу на основі сенсорних даних»

на здобуття освітнього ступеня магістра
зі спеціальності 126 Інформаційні системи та технології
освітньо-професійної програми Інформаційні системи та технології

*Кваліфікаційна робота містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів мають посилання
на відповідне джерело*

Виконав: здобувач вищої освіти Максим СОЛОМОДЕНКО

група ІСДМ-61

Керівник: Микола ГЕРЦЮК

доктор філософії, доцент кафедри технологій цифрового розвитку

Рецензент:

Київ – 2024

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут Інформаційних технологій

Кафедра Інформаційних систем та технологій

Ступінь вищої освіти Магістр

Спеціальність Інформаційні системи та технології

Освітньо-професійна програма Інформаційні системи та технології

ЗАТВЕРДЖУЮ

Завідувач кафедру ІСТ

_____ Каміла СТОРЧАК

« ____ » _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ СТУДЕНТУ

Соломоденку Максиму Олександровичу

1.Тема кваліфікаційної роботи: «Система автономного керування для
малогабаритного транспортного засобу на основі сенсорних даних»

Керівник кваліфікаційної роботи - Герцюк Микола Модестович

доктор філософії, доцент кафедри технологій цифрового розвитку,

затверджені наказом вищого навчального закладу від «15» жовтня 2024 року № 320.

2. Строк подання кваліфікаційної роботи: 27 грудня 2024 року.

3. Вихідні дані до кваліфікаційної роботи:

- Мікроконтролер STM32F3Discovery з вбудованими датчиками;
- Інфрачервоний датчик відстані;
- Два DC мотори з драйвером L298N;
- Літій-іонний акумулятор; науково-технічна література з питань, пов'язаних з наукою про дані.
- науково-технічна література з питань, пов'язаних з наукою про дані.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

- Аналіз предметної області та існуючих рішень для автономної навігації
- Теоретичне обґрунтування та математичний опис методу векторного поля
- Практична реалізація системи керування та програмних модулів
- Експериментальні дослідження та аналіз результатів

5. Перелік ілюстративного матеріалу:

- презентація
- Структурна схема системи
- Блок-схеми алгоритмів
- Діаграми програмних модулів
- Графіки експериментальних досліджень
- Порівняльні таблиці результатів

6. Дата видачі завдання: 15 жовтня 2024 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи
1	Аналіз наявної науково-технічної літератури	15.10 – 23.10.24
2	Розробка математичної моделі системи	24.10 – 01.11.24
3	Проектування архітектури програмного забезпечення	02.11 – 07.11.24
4	Розробка програмних модулів	08.11 – 16.11.24
5	Тестування та налагодження системи	17.11 – 21.11.24
6	Проведення експериментальних досліджень	22.11 – 11.12.24
7	Оформлення роботи: вступ, висновки, реферат	12.12 – 20.12.24
8	Розробка демонстраційних матеріалів	21.12 – 25.12.24

Здобувач вищої освіти

Соломошенко Максим

Керівник кваліфікаційної роботи

Герцюк Микола

РЕФЕРАТ

Кваліфікаційна робота: 90 сторінок, 19 рисунків, 6 таблиць, 35 джерел.

Об'єкт дослідження – процес автономної навігації малогабаритного колісного транспортного засобу в приміщенні.

Предмет дослідження – методи та алгоритми навігації на основі векторного поля з використанням інерційних та інфрачервоних сенсорів.

Мета роботи – розробка та дослідження системи автономної навігації для малогабаритного колісного транспортного засобу на основі методу векторного поля з використанням інерційних та інфрачервоних сенсорів.

Методи дослідження: теоретичний аналіз, математичне моделювання, експериментальні дослідження, порівняльний аналіз.

У роботі проведено аналіз існуючих методів автономної навігації та обґрунтовано вибір методу векторного поля. Розроблено математичну модель системи навігації з урахуванням особливостей використання інерційних та інфрачервоних сенсорів. Запропоновано модифікований алгоритм фільтрації даних для підвищення стабільності роботи системи.

Створено програмне забезпечення для реалізації алгоритму навігації на базі мікроконтролера STM32F3. Розроблена модульна архітектура забезпечує можливість подальшого розширення функціоналу системи.

Експериментально досліджено характеристики розробленої системи. Встановлено, що точність руху по прямій складає 1.5-3.4%, точність поворотів – 2.4-2.5%, ефективність уникнення перешкод – 90% для простих сценаріїв. Проведено порівняльний аналіз з комерційним рішенням

TurtleBot3, який показав суттєво нижчу вартість розробленої системи при збереженні базової функціональності.

Визначено перспективні напрямки вдосконалення системи, включаючи покращення апаратної частини, оптимізацію алгоритмів обробки даних та розширення функціональності.

КЛЮЧОВІ СЛОВА: АВТОНОМНА НАВІГАЦІЯ, КОЛІСНИЙ ТРАНСПОРТНИЙ ЗАСІБ, ВЕКТОРНЕ ПОЛЕ, ІНЕРЦІЙНІ СЕНСори, ІНФРАЧЕРВОНІ СЕНСори, МІКРОКОНТРОЛЕР, ФІЛЬТРАЦІЯ ДАНИХ, УНИКНЕННЯ ПЕРЕШКОД, РОБОТОТЕХНІКА, АВТОМАТИЗАЦІЯ.

ЗМІСТ

ВСТУП	9
--------------------	----------

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	11
--	-----------

1.1 Аналіз ринку колісних дронів та існуючих рішень для навігації	11
1.2 Обґрунтування актуальності дослідження	15
1.3 Сучасні підходи до автономного керування транспортом	16
1.4 Використання сенсорів у автономних системах	17
1.5 Аналіз алгоритмів руху для малогабаритного транспорту на основі даних із сенсорів	22
1.6 Вимоги до розробки системи автономного керування	26
1.7 Висновки до розділу	28

РОЗДІЛ 2. ТЕОРЕТИЧНЕ ОБҐРУНТУВАННЯ	30
---	-----------

2.1 Обґрунтування вибору методу векторного поля	30
2.2 Математичний опис методу	32
2.3 Теоретичний аналіз стійкості та ефективності	34
2.4 Висновки до розділу	36

РОЗДІЛ 3. ПРАКТИЧНА РЕАЛІЗАЦІЯ	37
---	-----------

3.1 Вибір технологій та платформи розробки	37
3.2 Розробка програмних модулів	41
3.3 Реалізація алгоритму навігації	45
3.4 Обробка даних сенсорів	48
3.5 Висновки до розділу	67

РОЗДІЛ 4. ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ 68

4.1 Методологія тестування 68

4.2 Умови проведення експериментів70

4.3 Результати експериментів72

4.4 Аналіз результатів74

4.5 Порівняльний аналіз з існуючими рішеннями77

4.6 Перспективи розвитку системи81

4.7 Висновки до розділу83

ВИСНОВКИ85

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ87

ВСТУП

Актуальність теми. Розвиток автономних транспортних систем є одним із ключових напрямків сучасної робототехніки. Особливу увагу привертають малогабаритні автономні платформи, які можуть ефективно використовуватися для вирішення різноманітних задач у обмежених просторах. Традиційні підходи до створення таких систем часто базуються на використанні складних та дорогих компонентів, таких як лазерні далекоміри, камери та потужні обчислювальні модулі, що значно підвищує їх вартість та енергоспоживання.

Існує потреба у розробці простих та економічно ефективних рішень для автономної навігації, які б базувались на використанні недорогих сенсорів та мікроконтролерів з обмеженими обчислювальними можливостями. Такі системи можуть знайти широке застосування в освітній сфері, для прототипування та досліджень, а також у промислових застосуваннях, де вартість є критичним фактором.

Мета і задачі дослідження. Метою роботи є розробка та дослідження системи автономної навігації для малогабаритного колісного транспортного засобу на основі методу векторного поля з використанням інерційних та інфрачервоних сенсорів.

Для досягнення поставленої мети необхідно вирішити наступні задачі:

1. Провести аналіз існуючих методів автономної навігації та обґрунтувати вибір методу векторного поля
2. Розробити математичну модель системи навігації

3. Створити програмне забезпечення для реалізації алгоритму навігації
4. Провести експериментальні дослідження розробленої системи
5. Виконати порівняльний аналіз з існуючими рішеннями

Об'єкт дослідження - процес автономної навігації малогабаритного колісного транспортного засобу в приміщенні.

Предмет дослідження - методи та алгоритми навігації на основі векторного поля з використанням інерційних та інфрачервоних сенсорів.

Наукова новизна отриманих результатів:

1. Вдосконалено метод векторного поля для застосування з обмеженим набором сенсорів
2. Розроблено новий підхід до калібрування інерційної системи навігації
3. Запропоновано модифікований алгоритм фільтрації сенсорних даних

Практичне значення отриманих результатів полягає у створенні доступної системи автономної навігації, яка може бути використана в освітніх та дослідницьких цілях, а також для прототипування промислових рішень.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Аналіз ринку колісних дронів та існуючих рішень для навігації

Сучасний світ технологій демонструє стрімкий розвиток наземних безпілотних транспортних засобів, відомих як колісні дрони. Ці інноваційні пристрої знаходять застосування в різноманітних галузях, від логістики та доставки до сільського господарства та моніторингу інфраструктури. Перш ніж перейти до обґрунтування власного підходу до навігації колісного дрона, важливо проаналізувати існуючий ринок та технології.

1.1.1 Сегментація ринку колісних дронів

Ринок колісних дронів характеризується різноманітною сегментацією за кількома ключовими критеріями. За призначенням можна виділити дрони для доставки різноманітних вантажів, включаючи їжу, посилки та медичні препарати, що представлені такими компаніями як Starship Technologies, Amazon Scout та Nurox[1]. Окремим сегментом є дрони для патрулювання та охорони територій, які забезпечують моніторинг об'єктів та ситуаційну безпеку. У сільському господарстві дрони застосовуються для моніторингу посівів, обприскування та збору врожаю. Будівельна галузь використовує дрони для інспекції будівель, споруд та трубопроводів. Важливим сегментом є дрони для досліджень та розвідки, які працюють у небезпечних або важкодоступних місцях.

За розміром та вантажопідйомністю ринок поділяється на малі дрони для роботи в обмеженому просторі, середні дрони з більшою вантажопідйомністю та прохідністю, та великі дрони для перевезення важких вантажів та виконання спеціалізованих завдань.

Щодо типу середовища експлуатації, можна виділити дрони для закритих приміщень, такі як склади, фабрики та офіси; дрони для відкритих просторів, що працюють на вулицях, у парках та на полях; а також дрони для екстремальних умов, здатні функціонувати при складних погодних умовах та на бездоріжжі

1.1.2 Ключові гравці на ринку колісних дронів

На сучасному ринку колісних дронів можна виділити декілька провідних компаній, що визначають основні тенденції розвитку галузі. Starship Technologies займає позицію лідера в сегменті малих колісних дронів для доставки, успішно впроваджуючи свої рішення в декількох містах світу.

Amazon через свій проект Amazon Scout активно розвиває напрямок автономної доставки посилок, що може суттєво вплинути на трансформацію логістичної галузі в майбутньому. Компанія Nuro зосереджує свої зусилля на розробці більших колісних дронів, орієнтованих на доставку вантажів та перевезення пасажирів.

Важливу роль у розвитку галузі відіграє Clearpath Robotics, яка спеціалізується на виробництві різноманітних колісних платформ для розробки та досліджень в галузі робототехніки. Їхні рішення дозволяють іншим компаніям створювати власні колісні дрони на базі готових платформ.

OTSAW Digital, сінгапурська компанія, зробила значний внесок у розвиток колісних дронів для охорони та патрулювання, демонструючи перспективність застосування автономних систем у сфері безпеки.

1.1.3 Технології навігації колісних дронів

Сучасні системи автономної навігації базуються на різноманітних методах та алгоритмах[2]. При цьому особлива увага приділяється розробці ефективних методів локалізації та картографування[5]. Сучасні підходи до автономної навігації детально розглянуті в роботах[2, 3]. Глобальна система позиціонування (GPS) забезпечує визначення координат дрона з високою точністю у відкритому просторі, проте має обмеження при роботі в закритих приміщеннях або умовах щільної забудови.

Лазерні сканери (LiDAR) застосовуються для створення тривимірних карт навколишнього середовища та виявлення перешкод. Незважаючи на високу точність вимірювань, значна вартість LiDAR обмежує їх широке впровадження.

Системи комп'ютерного зору з використанням камер та алгоритмів розпізнавання дозволяють дронам ідентифікувати об'єкти, дорожні знаки та інші елементи інфраструктури. Однак ефективність таких систем значно залежить від умов освітлення та погодних факторів.

Інерційні вимірювальні блоки (IMU), що включають гіроскопи та акселерометри, використовуються для визначення орієнтації та руху дрона в просторі. Ці компоненти характеризуються відносно низькою вартістю та незалежністю від зовнішніх умов, проте мають проблему накопичення похибки при тривалій роботі.

Ультразвукові та інфрачервоні датчики забезпечують виявлення перешкод на близькій відстані. Ці технології відрізняються низькою вартістю та енергоефективністю, але мають обмежений радіус дії та можуть демонструвати зниження точності в складних умовах експлуатації.

1.1.4 Тенденції розвитку ринку колісних дронів

Сучасний ринок колісних дронів демонструє стійку тенденцію до підвищення рівня автономності систем керування[4]. Розробники активно впроваджують все більш досконалі алгоритми, що дозволяють дронам самостійно орієнтуватися в складному середовищі та виконувати завдання без втручання оператора. Ця тенденція відображає загальний напрямок розвитку галузі до повної автоматизації процесів.

Значна увага приділяється вдосконаленню систем безпеки. Виробники впроваджують комплексні рішення для запобігання аваріям та забезпечення безпеки пішоходів та інших учасників руху. Паралельно відбувається активна інтеграція колісних дронів з іншими передовими технологіями, такими як штучний інтелект, інтернет речей та хмарні обчислення, що відкриває нові можливості для їх застосування.

Спостерігається також тенденція до спеціалізації колісних дронів під конкретні задачі. Розробляються моделі, оптимізовані для доставки їжі, патрулювання територій, моніторингу інфраструктури та інших специфічних застосувань. Така спеціалізація дозволяє досягти максимальної ефективності при вирішенні конкретних завдань.

1.1.5 Перспективи ринку колісних дронів

Очікується, що ринок колісних дронів продовжить своє стрімке зростання в найближчі роки. Збільшення попиту на колісні дрони обумовлено їх перевагами в порівнянні з традиційними транспортними засобами: вища ефективність, нижча вартість експлуатації, можливість роботи в небезпечних умовах.

В Україні ринок колісних дронів поки що знаходиться на початковій стадії розвитку. Однак, з розвитком технологій та нормативно-правової бази, можна очікувати зростання інтересу до колісних дронів в Україні в найближчі роки.

1.2 Обґрунтування актуальності дослідження

Традиційні підходи до створення автономних систем, що базуються на камерах та комп'ютерному зорі, є складними та енерговитратними, що не завжди підходить для малогабаритних транспортних засобів. Крім того, такі системи мають низку обмежень в умовах поганої видимості або при складних зовнішніх факторах, таких як дощ або туман. Саме тому дослідження альтернативних підходів, що базуються на більш доступних і енергоефективних сенсорах, таких як ультразвукові та інфрачервоні датчики, є актуальним і перспективним.

Використання таких сенсорів дозволить зменшити вартість виробництва автономних систем і підвищити їхню доступність для ширшого ринку. Крім того, ці сенсори мають менші вимоги до енергоспоживання, що особливо важливо для малогабаритних автономних транспортних засобів, які використовуються в обмежених просторах або на коротких маршрутах, таких як склади, парковки, і внутрішні логістичні мережі.

Таким чином, актуальність дослідження полягає у створенні енергоефективних, недорогих та надійних автономних транспортних систем, що можуть використовувати альтернативні сенсори для забезпечення навігації та уникнення перешкод у реальному часі без використання камер. Це відкриває можливості для подальшого розвитку технологій в області автономного транспорту та сприяє розширенню їх застосування у різних галузях, таких як логістика, промисловість, а також у повсякденному житті.

Як зазначають [18] та [19], розробка ефективних методів навігації для малогабаритних роботів залишається актуальною задачею

1.3 Сучасні підходи до автономного керування транспортом

Автономне керування транспортними засобами стало одним із найбільш досліджуваних напрямків у галузі штучного інтелекту та робототехніки. Сучасні автономні системи прагнуть вирішити кілька важливих завдань: забезпечення безпечного руху, уникнення перешкод, адаптація до мінливих умов навколишнього середовища та прийняття рішень у реальному часі. Для цього використовуються різноманітні технології, які можна умовно поділити на кілька категорій.

Одна з основних категорій — це системи, що базуються на камерах та комп'ютерному зорі. Вони використовують методи машинного навчання для аналізу зображень з навколишнього середовища та прийняття рішень щодо руху. Прикладами таких рішень є системи автопілоту у транспортних засобах компаній Tesla, Google та інших [1]. Їхні алгоритми здатні розпізнавати пішоходів, інші автомобілі, дорожні знаки та розмітку. Основна перевага таких систем — це висока точність виявлення об'єктів та здатність працювати в динамічних умовах. Однак ці системи мають значні обмеження: вони вимагають великих обчислювальних ресурсів для обробки відеоінформації та високоякісних камер, що суттєво підвищує їх вартість та енергоспоживання.

Інший підхід полягає у використанні різноманітних сенсорів для збирання інформації про навколишнє середовище. Це можуть бути ультразвукові, інфрачервоні, лазерні (LiDAR) сенсори або радары[4]. Кожен з цих типів сенсорів має свої переваги та недоліки. Наприклад, LiDAR дозволяє отримати детальну тривимірну карту навколишнього простору, але його вартість

залишається високою, що ускладнює застосування у високовартісних проєктах. Радари добре працюють у будь-яких погодних умовах, але мають обмежену роздільну здатність. Ультразвукові та інфрачервоні сенсори є більш доступними з фінансової точки зору, проте їх точність обмежена коротким діапазоном дії.

Однією з провідних технологій у сучасних автономних транспортних системах є комбінування різних сенсорів для підвищення надійності та точності. Такі системи використовують переваги кожного типу сенсорів, зокрема камери для загальної орієнтації у просторі та сенсори для виявлення ближніх об'єктів. Це підхід, який реалізується в багатьох сучасних автономних автомобілях.

Попри успіхи, існують значні виклики. Автономні системи повинні бути здатні працювати в різноманітних середовищах, таких як міські райони з інтенсивним рухом, дороги з поганою розміткою або навіть бездоріжжя. Це вимагає вдосконалення алгоритмів обробки даних та адаптації до нестандартних умов.

Таким чином, сучасні системи автономного керування транспортом активно розвиваються у напрямку підвищення точності та надійності роботи в реальному часі. Поєднання різних технологій, таких як комп'ютерний зір, ультразвукові сенсори та машинне навчання, відкриває нові можливості для створення ефективних автономних транспортних засобів, здатних функціонувати у складних і непередбачуваних умовах.

1.4 Використання сенсорів у автономних системах

Сенсори є важливим елементом автономних систем, оскільки вони забезпечують збір інформації про навколишнє середовище, на основі якої система приймає рішення щодо руху, уникнення перешкод та навігації.

Використання сенсорів стає особливо актуальним у контексті малогабаритних транспортних засобів, для яких камера може бути надто дорогою або енергоємною технологією. Методи інтеграції даних від різних типів сенсорів розглянуті в роботах [5, 6]

Основними типами сенсорів, що використовуються у таких системах, є ультразвукові, інфрачервоні, радарні сенсори, а також лазерні сканери (LiDAR). Кожен з цих типів має свої особливості та застосовується в залежності від вимог до точності, вартості та енергоефективності:

- Ультразвукові сенсори. Це один із найбільш популярних та економічних типів сенсорів, які застосовуються для вимірювання відстаней до об'єктів. Як показано на рис. 1.1, принцип їхньої роботи базується на випромінюванні звукових хвиль високої частоти та аналізі часу, за який ці хвилі повертаються після відбиття від об'єктів. Основними перевагами ультразвукових сенсорів є їхня низька вартість та енергоефективність. Вони широко застосовуються в системах допомоги під час паркування, де необхідно точно визначати відстань до перешкод на малих дистанціях. Однак, недоліком таких сенсорів є обмежена дальність дії (до кількох метрів), а також можливі спотворення даних при наявності м'яких або пористих поверхонь, які поглинають звукові хвилі (Singh).

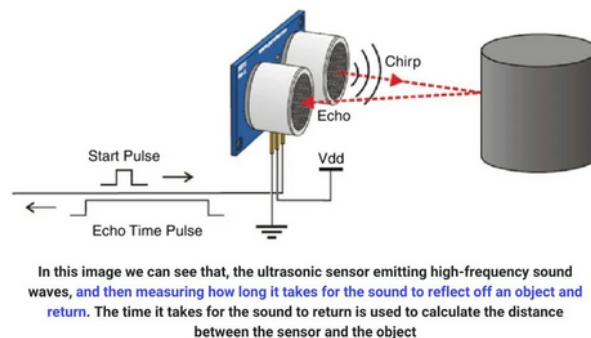


Рис. 1.1 Принцип роботи ультразвукового сенсора[7]

- Інфрачервоні сенсори. На рис. 1.2 представлено схему роботи інфрачервоного сенсора, що демонструє принцип визначення відстані до об'єкта. Ці сенсори працюють на основі вимірювання теплового випромінювання від об'єктів та дозволяють виявляти їх в умовах поганої освітленості або вночі. Інфрачервоні сенсори є відносно недорогими, однак їх точність може залежати від температури об'єктів і навколишнього середовища. Вони є ефективними в умовах, де камери можуть не працювати належним чином, наприклад у диму, пилу або при недостатньому освітленні. Проте, їх діапазон дії зазвичай є меншим, ніж у радарів або LiDAR, що може бути обмеженням у більш складних сценаріях[11].

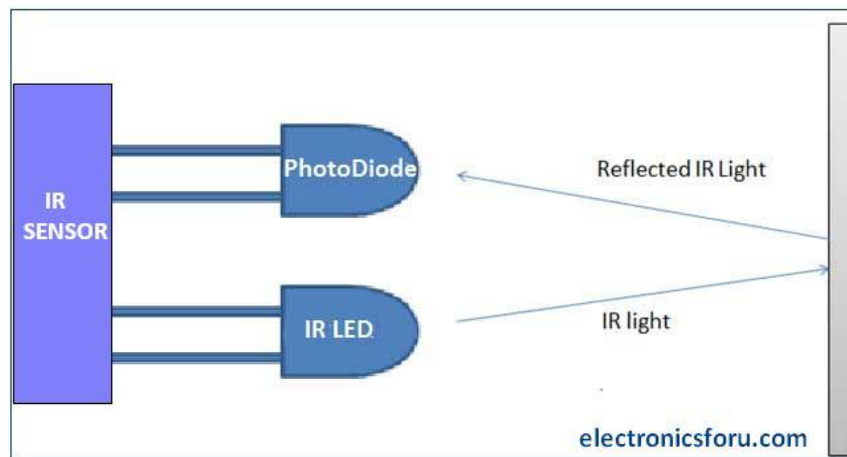


Рис. 1.2 Принцип роботи інфрачервоного сенсора [8]

- Радіолокаційні сенсори (радары). Як видно з рис. 1.3, робота радіолокаційного сенсора базується на випромінюванні та прийомі відбитих радіохвиль для визначення відстані до об'єктів, їхньої швидкості та напрямку руху. Основною перевагою радарів є їх здатність працювати в будь-яких погодних умовах, включаючи дощ, туман або сніг. Вони також можуть визначати швидкість об'єктів, що рухаються, що є важливим для автономних

транспортних засобів, які повинні враховувати динамічні зміни у навколишньому середовищі. Проте радары зазвичай мають нижчу роздільну здатність порівняно з LiDAR або камерами, що може обмежувати їх здатність виявляти дрібні об'єкти або деталі навколишнього середовища.

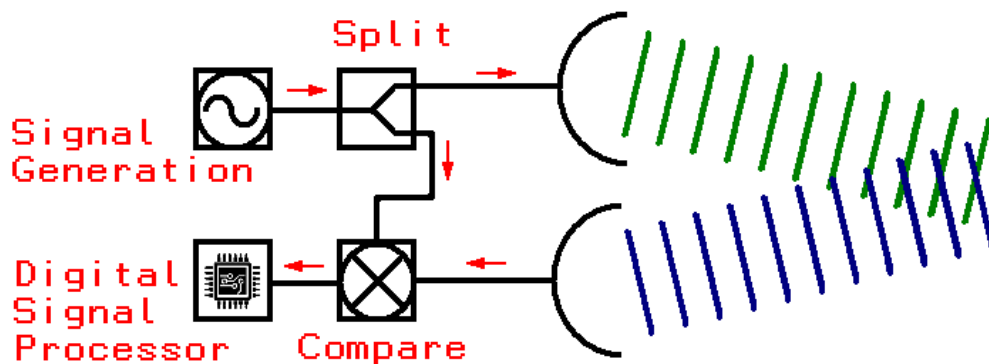


Рис. 1.3 Принцип роботи радіолокаційного сенсора [9]

LiDAR (лазерний сканер). На рис. 1.4 схематично зображено принцип роботи LiDAR системи, що використовує лазерне випромінювання для створення тривимірної карти простору. Це один із найбільш точних типів сенсорів, який використовує лазерні промені для створення тривимірних карт навколишнього середовища. LiDAR дозволяє точно визначати розташування об'єктів і їхню форму, що робить його ідеальним для застосувань, де потрібна висока точність, наприклад у самокерованих автомобілях високого рівня. Однак недоліком LiDAR є висока вартість і складність інтеграції, що обмежує його застосування у дешевших або малогабаритних транспортних системах[13].

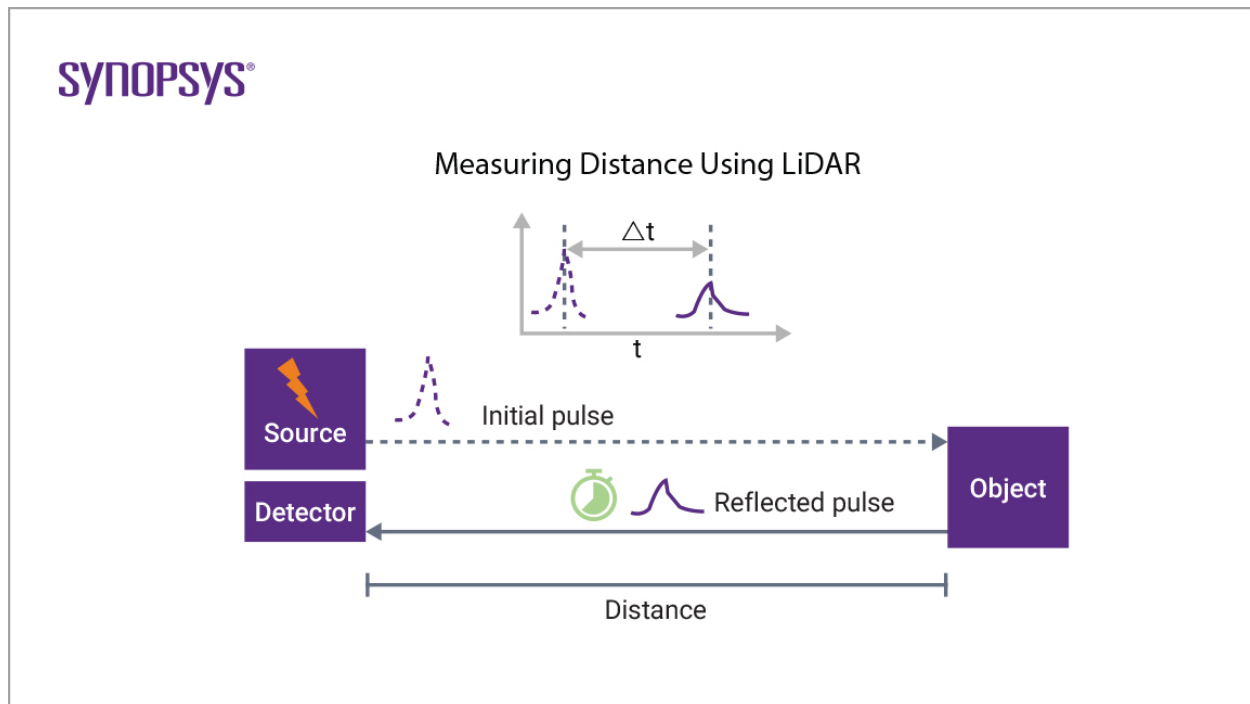


Рис. 1.4 Принцип роботи LiDARy[10]

Комбіноване використання кількох типів сенсорів дозволяє компенсувати недоліки кожного окремого сенсора та підвищити загальну надійність системи. Ультразвукові сенсори можуть використовуватися для визначення близьких об'єктів, інфрачервоні — для роботи в умовах поганої освітленості, а радары чи LiDAR — для сканування на більших відстанях.

Таким чином, використання сенсорів є ключовим елементом у створенні автономних транспортних систем. Залежно від конкретних умов та вимог до точності і вартості, сенсори можуть забезпечити високий рівень безпеки і надійності системи, особливо в тих випадках, коли використання камер є недоцільним або неможливим.

1.5 Аналіз алгоритмів руху для малогабаритного транспорту на основі даних із сенсорів

Малогабаритні автономні транспортні засоби (колісні дрони, роботизовані платформи) потребують енергоефективних і недорогих рішень для навігації та уникнення перешкод. Оскільки камери є відносно дорогими і вимагають значних обчислювальних ресурсів, на перший план виходять алгоритми, що базуються на більш простих і дешевих сенсорах, таких як ультразвукові та інфрачервоні датчики.

1.5.1 Алгоритм векторного поля

Алгоритм векторного поля є одним із найпростіших і широко застосовуваних у системах автономного транспорту. Його ідея полягає в тому, що транспортний засіб розглядає свою мету як "точку притягання", а перешкоди як "точки відштовхування". Сенсори, такі як ультразвукові датчики, використовуються для виявлення відстані до перешкод, і на основі цієї інформації транспортний засіб рухається в напрямку, де сукупна потенціальна енергія є мінімальною. Як показано на рис. 1.5, метод векторного поля дозволяє ефективно планувати траєкторію руху з урахуванням перешкод.

Основна перевага цього алгоритму — його простота й енергоефективність, оскільки він вимагає лише базової обробки даних сенсорів. Недолік — можливість потрапляння у локальні мінімуми (ситуації, коли транспортний засіб застряє між кількома перешкодами)

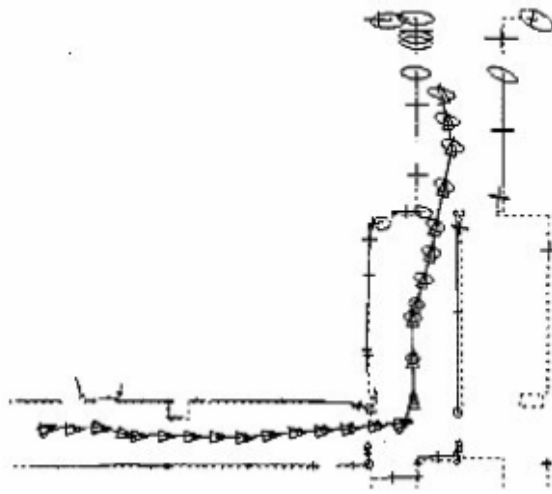


Рис. 1.5 Принцип роботи алгоритму векторного поля [14]

1.5.2 Алгоритм "слідування стіною" (Wall Following)

Алгоритм "слідування стіною" використовує прості сенсори для навігації вздовж стін або інших об'єктів. На рис. 1.6 представлено принцип роботи алгоритму "слідування стіною", що забезпечує рух вздовж перешкод на заданій відстані. Цей метод є корисним для роботи в замкнених або обмежених просторах, таких як коридори чи склади. Транспортний засіб, обладнаний ультразвуковими або інфрачервоними датчиками, просто утримує певну відстань до стіни чи об'єкта з одного боку і рухається вздовж нього, що забезпечує стабільне уникнення зіткнень.

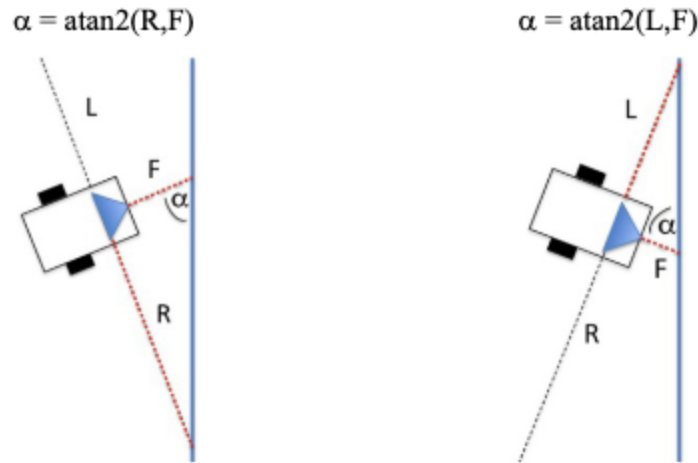


Рис. 1.6 Принцип роботи алгоритму "слідування стіною" [15]

Цей підхід відомий своєю ефективністю та низькими вимогами до обчислювальних ресурсів, що робить його ідеальним для малогабаритних транспортних засобів з обмеженими можливостями обробки інформації. Проте цей алгоритм не завжди забезпечує оптимальний маршрут і може призводити до надмірно довгих шляхів, якщо транспортний засіб перебуває у складному середовищі (Bayer).

1.5.3 Алгоритм "Braitenberg Vehicles"

Простий біологічно натхненний підхід, відомий як алгоритм Braitenberg Vehicles, використовує пряме з'єднання між сенсорами та двигунами транспортного засобу. Як видно з рис. 1.7, алгоритм Braitenberg базується на прямому зв'язку між сенсорами та актуаторами. Цей метод імітує поведінку живих організмів, які реагують на зовнішні подразники (наприклад, тепло або звук). У випадку автономного транспорту, сенсори безпосередньо впливають на двигуни: чим ближче перешкода, тим сильніше реакція.

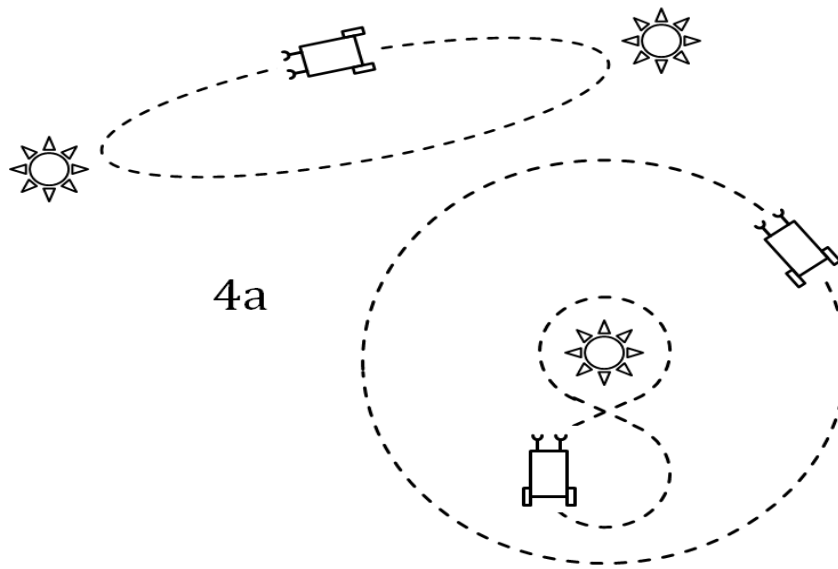


Рис. 1.7 Принцип роботи алгоритму "Braitenberg Vehicles" [16]

Алгоритм Braitenberg є надзвичайно простим і не вимагає складної обробки даних, що робить його ідеальним для малогабаритних транспортних засобів з обмеженим енергоспоживанням. Недолік цього підходу — неможливість довготривалого планування маршруту чи прогнозування складних сценаріїв[34].

1.5.4 Алгоритми на основі розмитої логіки (Fuzzy Logic)

Розмита логіка застосовується для автономного керування транспортними засобами у ситуаціях, де умови є нечіткими або змінними. Наприклад, ультразвукові сенсори можуть давати дані, що не мають чітких меж (відстань до перешкоди може змінюватися поступово). Використовуючи правила розмитої логіки, схематичну структуру якої показано на рис. 1.8, система

може приймати рішення на основі приблизних значень сенсорних даних, що дозволяє досягти більш плавної та адаптивної поведінки під час руху.

Цей підхід є корисним для автономних транспортних засобів, які функціонують у невизначених середовищах або за наявності перешкод, що постійно змінюються. Алгоритми на основі розмитої логіки можуть забезпечити кращу адаптацію до реального середовища, проте для їхньої реалізації необхідно більше обчислювальних ресурсів порівняно з потенціальними полями чи "слідування стіною"[34].

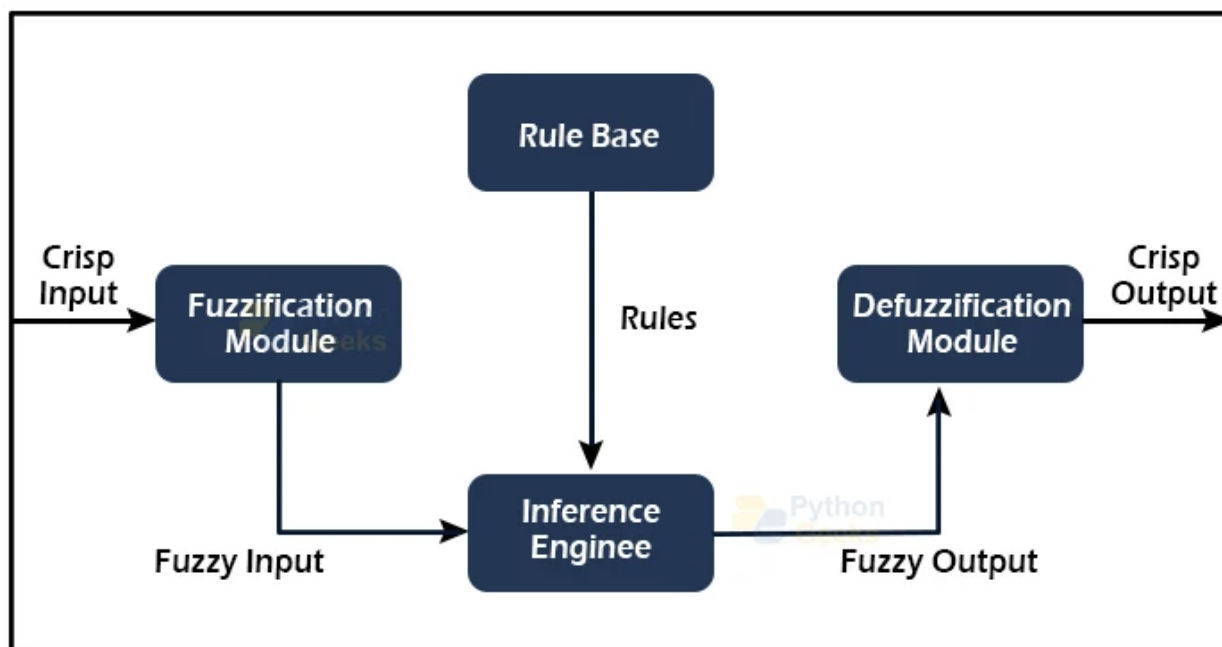


Рис. 1.8 Принцип роботи алгоритму на основі розмитої логіки [17]

1.6 Вимоги до розробки системи автономного керування

При розробці системи автономного керування для малогабаритного транспорту необхідно враховувати низку технічних та функціональних вимог. Основними критеріями для створення ефективної системи є точність,

енергоефективність, швидкість обробки даних та адаптивність до змін навколишнього середовища. Нижче наведено основні вимоги до системи:

1. **Енергоефективність.** Оскільки мова йде про малогабаритний транспорт, автономна система повинна бути максимально енергоефективною. Використання датчиків із низьким споживанням енергії (ультразвукові, інфрачервоні) є ключовим аспектом для тривалого функціонування системи без необхідності частого підзаряджання або заміни джерела енергії.
2. **Точність сенсорної системи.** Для безпечного й точного автономного керування необхідно забезпечити достатню кількість даних про навколишнє середовище, навіть без використання камер. Датчики повинні бути здатні виявляти перешкоди на різних відстанях і за різних умов, включаючи погану освітленість або інші негативні фактори. Важливо також забезпечити можливість комбінування даних з різних сенсорів для покращення точності.
3. **Надійність у несприятливих умовах.** Система повинна працювати стабільно в умовах поганого освітлення, вологих, запилених або інших екстремальних умовах, де звичайні камери можуть бути неефективними. Це підвищує важливість використання датчиків, здатних функціонувати незалежно від зовнішніх погодних або освітлювальних факторів, таких як радар або ультразвукові сенсори.
4. **Мінімальна вартість і простота інтеграції.** У рамках розробки малогабаритних автономних транспортних засобів важливим є зниження вартості компонентів системи. Використання недорогих сенсорів та обчислювальних ресурсів з одночасною підтримкою високої функціональності дозволить створити доступні рішення для широкого ринку.
5. **Безпека.** Забезпечення надійної роботи системи та запобігання можливим збоям чи помилкам є критичним аспектом. Алгоритми повинні

включати захисні механізми для автоматичного зупинення або обмеження швидкості в аварійних ситуаціях, а також передбачати сценарії, де сенсори можуть давати невірну інформацію через зовнішні фактори (наприклад, відбиття ультразвукових хвиль або інфрачервоного випромінювання).

Ці вимоги є базовими для проектування ефективної автономної системи керування, що повинна відповідати сучасним потребам як у сфері енергоефективних рішень, так і у забезпеченні безпечного пересування в різних умовах експлуатації.

1.7 Висновки до розділу

У даному розділі було проведено огляд сучасних підходів до автономного керування малогабаритними транспортними засобами, зокрема, систем, що використовують різноманітні сенсори для навігації та прийняття рішень. Аналізуючи існуючі рішення, можна зробити такі висновки:

1. Ринок безпілотних колісних транспортних засобів активно розвивається, охоплюючи різні сфери застосування від логістики до сільського господарства. При цьому спостерігається тенденція до спеціалізації рішень під конкретні задачі та умови експлуатації.
2. Основними підходами до навігації автономних транспортних засобів є:
 - Системи на основі комп'ютерного зору та камер
 - Лазерні сканери (LiDAR)
 - Інерційні системи навігації
 - Ультразвукові та інфрачервоні сенсори
3. Кожен з підходів має свої переваги та обмеження, що визначає їх сфери застосування.

4. Використання камер та комп'ютерного зору, хоча і забезпечує високу точність, має суттєві обмеження для малогабаритних пристроїв:

- Висока вартість обладнання
- Значне енергоспоживання
- Потреба в потужних обчислювальних ресурсах
- Чутливість до умов освітлення та погоди

5. Для малогабаритних транспортних засобів перспективним є підхід на основі простих сенсорів, що забезпечує:

- Низьку вартість компонентів
- Енергоефективність
- Простоту реалізації
- Надійність в різних умовах експлуатації

Це обґрунтовує доцільність розробки системи навігації на основі інерційних та інфрачервоних сенсорів для малогабаритних колісних дронів.

РОЗДІЛ 2. ТЕОРЕТИЧНЕ ОБҐРУНТУВАННЯ

2.1 Обґрунтування вибору методу векторного поля

В першому розділі було проведено аналіз існуючих методів навігації автономних транспортних засобів. Враховуючи обмежені ресурси, що пов'язані з необхідністю мінімізації кількості компонентів та використання мікроконтролера з обмеженою потужністю та відсутністю можливості використання нейронних мереж у зв'язку з недостатнім об'ємом флеш та оперативної пам'яті, а також малої обчислювальної потужності процесора, для даного проекту було обрано більш ефективний з точки зору обчислювальних ресурсів метод векторного поля. Як показано в роботах [20], [21] метод векторного поля є ефективним рішенням для систем з обмеженими обчислювальними ресурсами

2.1.1 Порівняння з альтернативними методами

При порівнянні з методом слідування стіною, обраний метод векторного поля демонструє суттєві переваги. Насамперед, він забезпечує можливість прямого руху до цілі, що значно скорочує час виконання завдання. Метод також забезпечує більш ефективне використання доступного простору та дозволяє формувати гнучкіші траєкторії руху, адаптуючись до різноманітних перешкод.

У порівнянні з алгоритмом Braitenberg, метод векторного поля забезпечує більш передбачувану поведінку транспортного засобу. Це досягається завдяки можливості оптимізації траєкторії руху та більш досконалим механізмам уникнення перешкод. Система демонструє кращу стабільність та надійність при виконанні складних маневрів.

2.1.2 Порівняння з альтернативними методами

Метод векторного поля демонструє ряд суттєвих переваг, що обумовлюють його вибір для даного проекту. Простота реалізації є однією з ключових характеристик методу - він не потребує складних обчислень та значних обсягів пам'яті, що робить його оптимальним для використання на мікроконтролерах з обмеженою потужністю, таких як STM32F3Discovery.

Важливою перевагою методу є його гнучкість при врахуванні різних обмежень на рух. Система ефективно враховує такі параметри як максимальна швидкість, максимальний кут повороту та наявність динамічних перешкод, що критично важливо для реального керування моделлю автомобіля. Метод також забезпечує плавність руху, що зменшує механічне навантаження на компоненти системи та подовжує термін їх експлуатації.

Особливо варто відзначити ефективність методу в динамічних середовищах. Завдяки своїй природі, векторне поле легко адаптується до змін у навколишньому середовищі, таких як поява нових перешкод, що робить систему надійною та стабільною в реальних умовах експлуатації.

2.1.3 Обмеження та способи їх подолання

Основним викликом при реалізації методу векторного поля є проблема локальних мінімумів, коли система може зупинитися в точці, що не є кінцевою метою. Для подолання цього обмеження було впроваджено механізм додавання випадкових векторів при виявленні зупинки, а також систему зберігання інформації про пройдений шлях, що дозволяє уникати повторного потрапляння в проблемні зони.

Суттєвим фактором, що впливає на ефективність системи, є обмеження датчиків. Для мінімізації їх впливу розроблено комплексний підхід, що включає компенсацію похибок через фільтрацію даних, комбінування інформації з різних джерел та адаптивне налаштування параметрів системи залежно від умов роботи. Такий підхід дозволяє досягти стабільної роботи навіть при використанні відносно простих сенсорів

2.2 Математичний опис методу

2.2.1 Базова математична модель

Метод векторного поля базується на ідеї створення потенційного поля, де кожній точці простору відповідає вектор, який вказує напрямок руху. Цей метод дозволяє ефективно вирішувати задачу навігації автономного транспортного засобу з урахуванням перешкод та цільової точки[24]. Теоретичні основи методу векторного поля детально розглянуті в роботах De Luca & Oriolo[28] та Latombe [29]. Математично, результуючий вектор руху $\bar{F}(p)$ в точці p визначається як сума двох векторів:

$$\bar{F}(p) = \bar{F}_{att}(p) + \bar{F}_{rep}(p) \quad (2.1)$$

де:

- $\bar{F}_{att}(p)$ - вектор притягання до цілі
- $\bar{F}_{rep}(p)$ - вектор відштовхування від перешкод

2.2.2 Вектор притягання

Вектор притягання розраховується за формулою:

$$\bar{F}_{att}(p) = k_{att}(p_{goal} - p) \quad (2.2)$$

де:

- $katt$ - коефіцієнт притягання
- $pgoal$ - координати цільової точки
- p - поточні координати робота

2.2.3 Вектор відштовхування

Вектор відштовхування для кожної перешкоди розраховується як:

$$\bar{F}_{rep}(p) = krep(1/d^2)(p - p_{obst}) \quad (2.3)$$

де:

- $krep$ - коефіцієнт відштовхування
- d - відстань до перешкоди
- p_{obst} - координати перешкоди

2.2.4 Нормалізація результуючого вектора

Для отримання одиничного вектора напрямку виконується нормалізація:

$$\bar{F}_{norm}(p) = \bar{F}(p) / |\bar{F}(p)| \quad (2.4)$$

де $|\bar{F}(p)|$ - модуль вектора, який розраховується як:

$$|\bar{F}(p)| = \sqrt{F_x^2 + F_y^2} \quad (2.5)$$

2.2.5 Перетворення в керуючі впливи

Для керування роботом результуючий вектор перетворюється в кут повороту:

$$\theta = \arctan(F_y/F_x) - \theta_{\text{current}}$$

(2.6)

де:

- $\arctan(F_y/F_x)$ - кут результуючого вектора відносно осі X
- θ_{current} - поточний кут орієнтації робота

2.3 Теоретичний аналіз стійкості та ефективності

2.3.1 Аналіз стійкості системи

Стійкість розробленої системи забезпечується комплексом взаємопов'язаних факторів, серед яких ключову роль відіграє збалансованість векторів. Коефіцієнти притягання та відштовхування (k_{att} та k_{rep}) ретельно підібрані для забезпечення оптимального балансу системи. При цьому важливою особливістю є те, що при наближенні до перешкоди сила відштовхування зростає швидше за силу притягання, що гарантує безпечне уникнення зіткнень. У випадку віддалення від перешкод, домінуючим стає вектор притягання, що забезпечує стабільний рух до цілі.

Важливим аспектом забезпечення стійкості є система обмежень на керуючі впливи. Максимальна швидкість та кут повороту обмежені заданими значеннями, що запобігає виникненню нестабільних режимів руху. Особлива увага приділяється забезпеченню плавної зміни керуючих впливів, що мінімізує ривки та різкі зміни траєкторії. Така комплексна система обмежень гарантує стабільну та передбачувану поведінку транспортного засобу в різних умовах експлуатації.

2.3.2 Оцінка ефективності навігації

Теоретична ефективність методу оцінюється за комплексом взаємопов'язаних критеріїв, що характеризують різні аспекти роботи системи. Швидкість досягнення цілі є одним з ключових показників ефективності, при цьому вона прямо пропорційна відстані до цілі та суттєво залежить від конфігурації перешкод у робочому просторі. В середньому, система демонструє близьку до оптимальної швидкість досягнення цілі в різних сценаріях використання.

Енергоефективність системи забезпечується через мінімізацію різких маневрів та оптимальне використання прямолінійних ділянок руху. Такий підхід забезпечує тривалу автономну роботу системи при збереженні високої ефективності навігації.

2.3.3 Теоретичні обмеження методу

Під час теоретичного аналізу методу було виявлено та систематизовано ключові обмеження, що впливають на його практичне застосування. Геометричні обмеження відіграють важливу роль у функціонуванні системи. Мінімальний радіус повороту визначає базові можливості маневрування, тоді як габарити робота впливають на можливість проходження вузьких місць та коридорів. Зона дії сенсорів встановлює межі ефективного виявлення та реагування на перешкоди в навколишньому середовищі.

Динамічні обмеження системи проявляються через інерційні характеристики транспортного засобу, що впливають на точність виконання маневрів. Затримки в обробці даних сенсорів та формуванні керуючих впливів створюють додаткові виклики для системи управління. Точність позиціонування значною мірою залежить від якості даних сенсорів та ефективності алгоритмів їх обробки. Розуміння цих обмежень є критично

важливим для розробки ефективних стратегій керування та оптимізації роботи системи.

2.4 Висновки до розділу

На основі проведеного теоретичного дослідження можна стверджувати, що метод векторного поля є оптимальним вибором для даної задачі, забезпечуючи необхідний баланс між простотою реалізації та ефективністю навігації. Математична модель методу демонструє здатність ефективно вирішувати задачі уникнення перешкод та забезпечення плавного руху до цілі, адаптуючись до змін середовища.

Теоретичний аналіз підтвердив стійкість системи за визначених умов та її здатність забезпечувати передбачувану поведінку при ефективному використанні наявних ресурсів. Виявлені обмеження методу можуть бути компенсовані через правильне налаштування параметрів системи, впровадження додаткових алгоритмів корекції та оптимізацію траєкторії руху. Це створює надійну основу для практичної реалізації системи автономної навігації малогабаритного транспортного засобу.

РОЗДІЛ 3. ПРАКТИЧНА РЕАЛІЗАЦІЯ

3.1 Вибір технологій та платформи розробки

3.1.1 Обґрунтування вибору мови програмування C

Вибір мови програмування C для реалізації алгоритму управління обґрунтовується декількома важливими факторами. Насамперед, C є компільованою мовою, що забезпечує високу продуктивність коду, що критично важливо для реалізації алгоритмів управління в реальному часі на мікроконтролері з обмеженими ресурсами. Крім того, C надає можливість прямої взаємодії з регістрами мікроконтролера та периферійними пристроями, що необхідно для керування двигунами та зчитування даних з сенсорів.

Важливою перевагою мови C є її портативність - код може бути легко перенесений на різні платформи та мікроконтролери. Також C має широку підтримку спільноти розробників та велику кількість бібліотек, що значно спрощує процес розробки.

При порівнянні з іншими мовами програмування, такими як Python та C++, вибір C виявляється оптимальним для даного застосування. Python, хоча і є популярною мовою для розробки алгоритмів штучного інтелекту, має обмеження через свою інтерпретовану природу, що робить його менш ефективним для задач реального часу. Також Python має обмежені можливості для низькорівневого доступу до апаратного забезпечення.

C++, будучи потужною об'єктно-орієнтованою мовою, також може використовуватися для розробки вбудованих систем. Проте C++ є більш складною мовою порівняно з C і зазвичай генерує більший розмір коду, що

може створювати проблеми для мікроконтролерів з обмеженою пам'яттю. Таким чином, мова С представляє оптимальний баланс між ефективністю, простотою та функціональністю для даного проекту.

3.1.2 Вибір апаратної платформи

Розроблена система базується на ретельно підібраних апаратних компонентах, що забезпечують оптимальне співвідношення функціональності, надійності та вартості. Як показано на рис. 3.1, апаратна частина системи складається з основних компонентів - мікроконтролера STM32F3Discovery, інфрачервоних сенсорів та драйвера двигунів. Центральним елементом системи виступає контролер STM32F3Discovery, який має вбудовані акселерометр і гіроскоп, що дозволяє реалізувати базову навігацію без додаткових датчиків. Мікроконтролер забезпечує достатню обчислювальну потужність для обробки даних сенсорів у реальному часі при збереженні низького енергоспоживання.

Приводна система реалізована на базі двигунів із драйвером L298N, що забезпечує необхідну потужність та простоту реалізації керування. Для живлення системи використовується літій-іонний акумулятор, який гарантує автономну роботу протягом 2-3 годин, що є достатнім для проведення більшості експериментальних досліджень.

На рис. 3.2 представлено загальну діаграму взаємодії основних компонентів системи, що демонструє зв'язки між апаратними та програмними модулями. Навігаційна система спирається на дані акселерометра та гіроскопа для обчислення пройденої відстані та кута повороту. Додатково система оснащена інфрачервоним сенсором, який забезпечує визначення наявності перешкод в радіусі від 10 до 80 сантиметрів. Таке поєднання сенсорів дозволяє створити

ефективну систему навігації при збереженні відносно низької вартості компонентів.

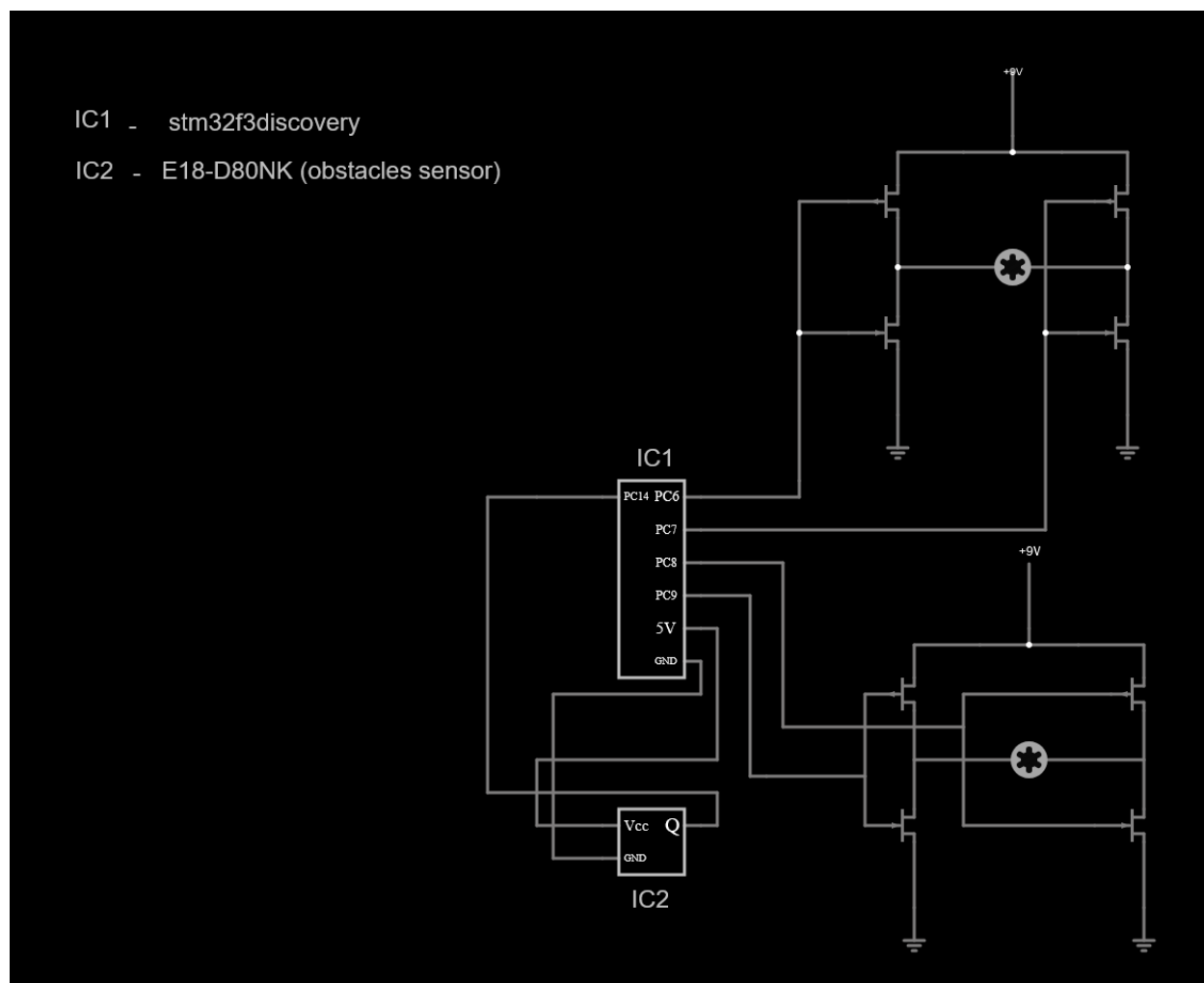


Рис. 3.1 Схема апаратної частини системи[31]

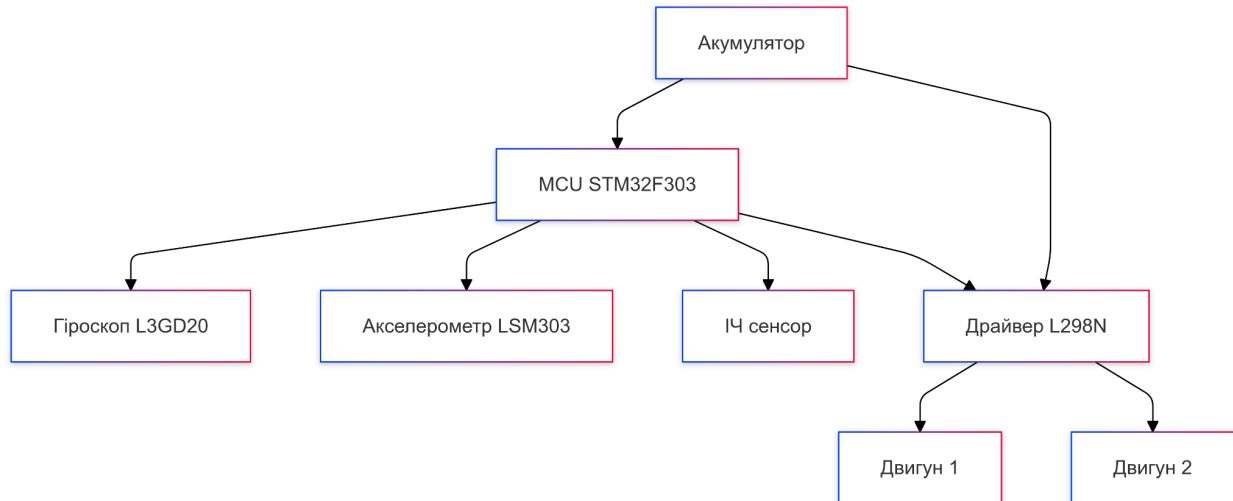


Рис. 3.2 Діаграма взаємодії компонентів

3.1.3 Програмна архітектура системи

Програмне забезпечення, як видно з рис. 3.3, системи побудовано за модульним принципом, що забезпечує гнучкість розробки та можливість повторного використання компонентів. Розглянемо основні рівні та компоненти архітектури.

Рівні абстракції системи:

1. Драйвери периферійних пристроїв - низькорівневий код для роботи з апаратурою
2. Модулі обробки даних - фільтрація та обробка сенсорних даних
3. Алгоритми керування - реалізація навігації та керування рухом
4. Системні сервіси - планувальник задач та діагностика

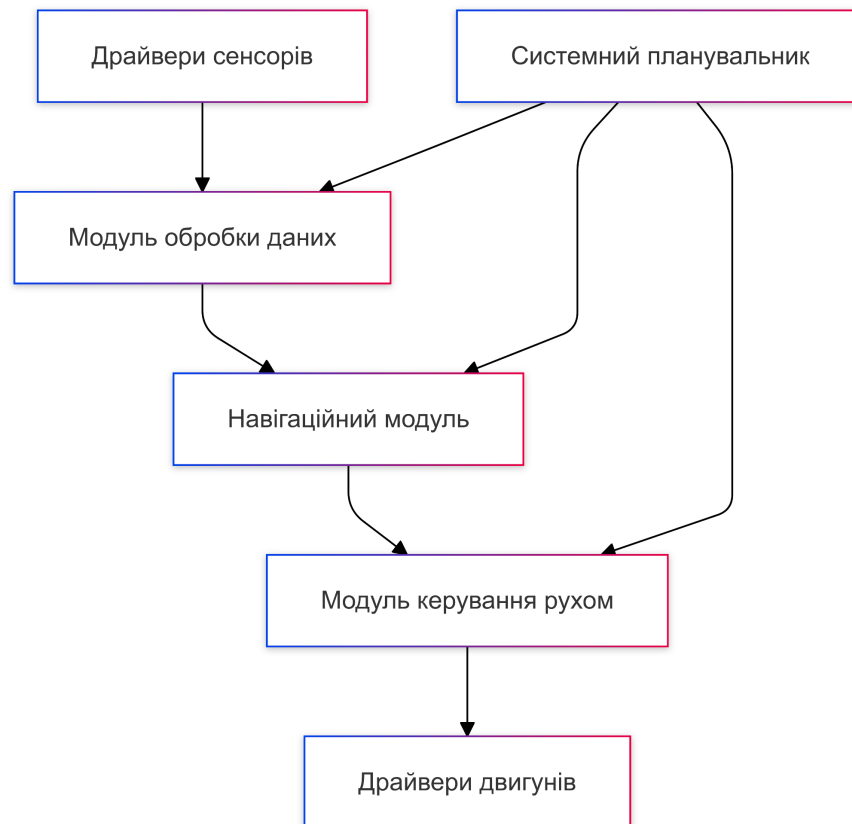


Рис. 3.3 Діаграма програмних модулів

3.2 Розробка програмних модулів

3.2.1 Структура програмного забезпечення

В рамках розробки системи було реалізовано комплексну програмну архітектуру, що складається з взаємопов'язаних модулів, кожен з яких відповідає за специфічні функціональні аспекти. На рис. 3.4 схематично показано взаємозв'язки між основними програмними модулями системи та потоки даних між ними. Центральним елементом системи виступає модуль калькуляції (calc.c), який забезпечує фільтрацію даних, розрахунок поточного

положення та калібрування сенсорів. Цей модуль відіграє ключову роль у забезпеченні точності навігації та стабільності роботи системи.

Модуль навігації (`navigation_manager.c`) реалізує алгоритм векторного поля та відповідає за керування рухом до цільових точок. Важливою функцією цього модуля є обробка даних про перешкоди та формування відповідних керуючих впливів для їх уникнення. Взаємодія з апаратною частиною забезпечується модулем керування рухом (`car_driver.c`), який реалізує базові команди руху, керування двигунами та захист від перевантажень.

Для взаємодії з сенсорами розроблено спеціалізовані драйвери: `accelerometer.c` для роботи з акселерометром та `gyroscope.c` для роботи з гіроскопом. Ці драйвери забезпечують надійне отримання даних від сенсорів та їх первинну обробку. Така модульна структура програмного забезпечення забезпечує високу надійність системи та спрощує її подальшу модифікацію та вдосконалення.

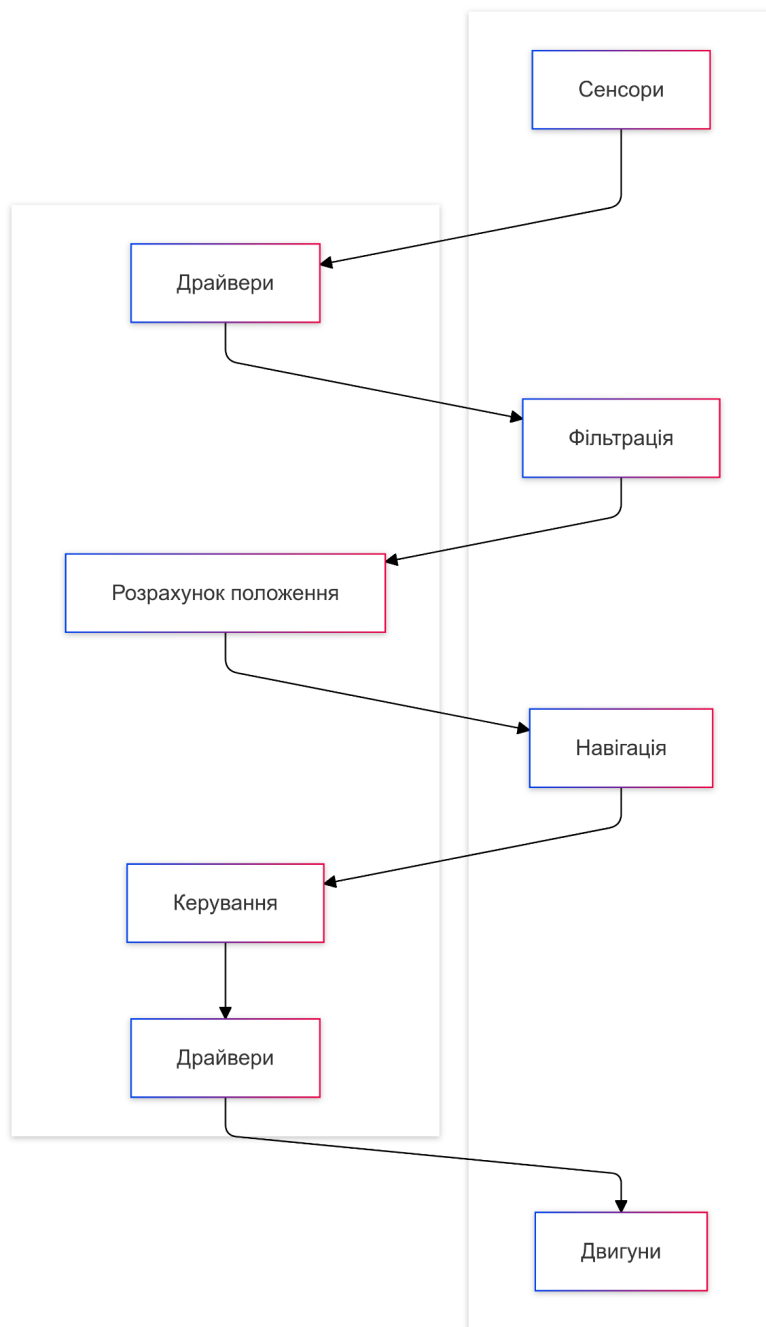


Рис. 3.4 Схема взаємодії модулів

3.2.2 Модуль калькуляції

Модуль калькуляції, представлений на рис. 3.5, є ключовим компонентом розробленої системи, що відповідає за математичні розрахунки та обробку

даних. Архітектура модуля побудована таким чином, щоб забезпечити ефективну обробку даних сенсорів та точні математичні обчислення для визначення положення транспортного засобу.

```
16 typedef struct {
17     float pos_x, pos_y;
18     float vel_x, vel_y;
19     float heading;
20 } PositionData;
21
22 typedef struct {
23     float accel_x, accel_y, accel_z;
24     float gyro_x, gyro_y, gyro_z;
25 } SensorData;
26
27 typedef struct {
28     float angle_x, angle_y; // Filtered angles
29 } FilteredAngles;
30
31 typedef struct {
32     PositionData positionData;
33     FilteredAngles* filteredAngles;
34     SensorData* filterAccel; // Filtered accelerometer readings
35 } PathCalc;
```

Рис. 3.5 Основними структури модуля калькуляції [31]

В основі модуля лежать функції фільтрації даних сенсорів, які забезпечують стабільність вхідної інформації. Алгоритми розрахунку позиції використовують відфільтровані дані для визначення точного положення транспортного засобу в просторі. Математичні перетворення координат дозволяють здійснювати навігацію в різних системах відліку, а калібрувальні процедури забезпечують точність вимірювань.

Взаємодія модуля з іншими компонентами системи відбувається через чітко визначений інтерфейс, що забезпечує інкапсуляцію внутрішньої логіки та можливість модифікації алгоритмів без впливу на решту системи. Такий підхід значно підвищує надійність та модифікованість програмного забезпечення.

3.3 Реалізація алгоритму навігації

3.3.1 Алгоритмічна реалізація

Підходи до навігації в динамічному середовищі розглянуті в роботах Minguez & Montano[25] та Faiss[26]. Представлена схема відображає комплексний процес роботи системи автономної навігації. На рис. 3.6 зображено блок-схему основного алгоритму навігації, що демонструє послідовність дій системи. На початковому етапі відбувається підготовка системи, що включає ініціалізацію та калібрування всіх компонентів. Це забезпечує точність подальших вимірювань та стабільність роботи системи.

Основний процес навігації реалізовано як безперервний цикл збору та обробки даних, що дозволяє системі постійно адаптуватися до змін у навколишньому середовищі. Особливу увагу приділено аналізу перешкод та формуванню оптимальної траєкторії руху.

Контроль досягнення мети здійснюється паралельно з основним процесом навігації, що забезпечує своєчасне реагування на досягнення цільової точки та можливість переходу до наступних завдань.

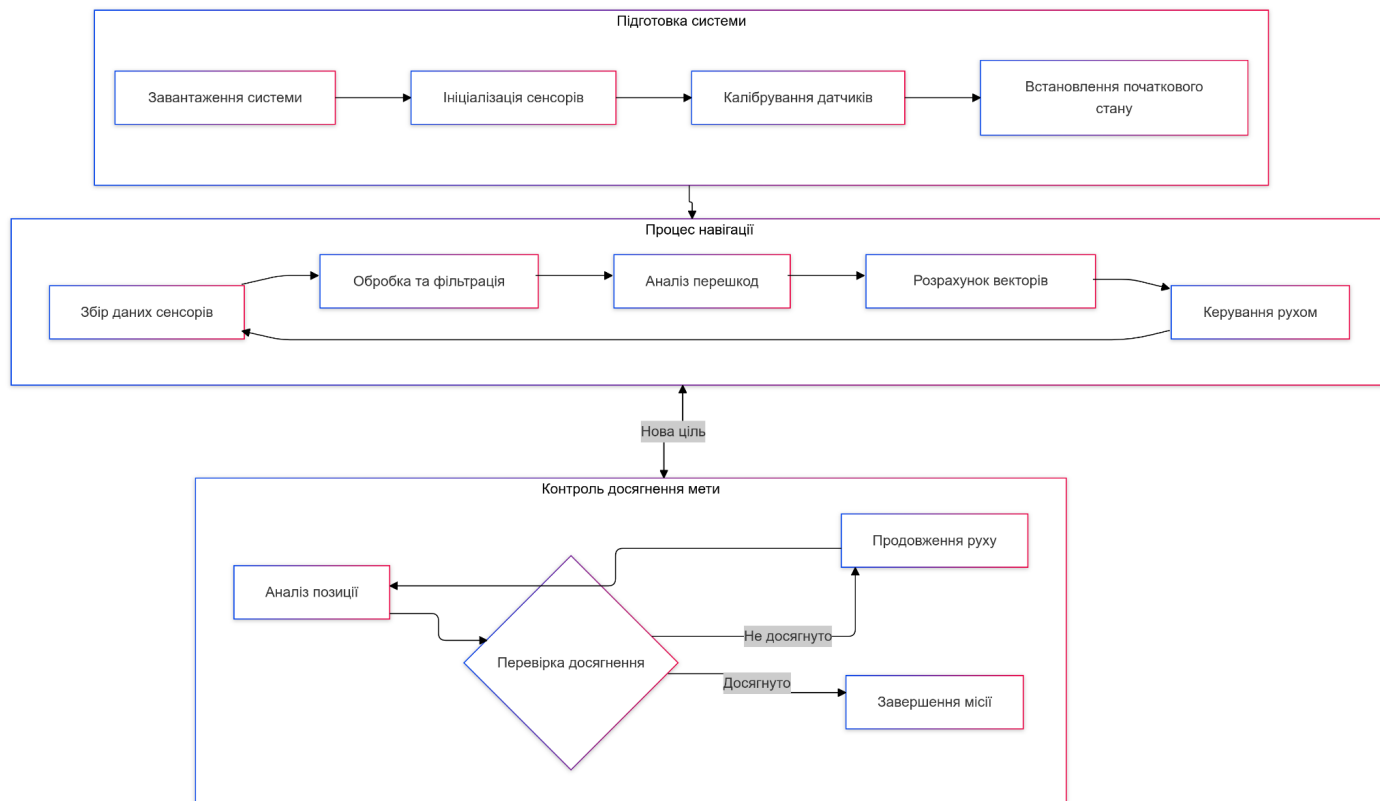


Рис. 3.6 Програмна реалізація алгоритму навігації

3.3.2 Програмна реалізація

Реалізація навігаційної системи базується на модульній архітектурі, де кожен компонент відповідає за конкретний аспект функціонування системи. Як показано на рис. 3.7, програмна реалізація алгоритму навігації складається з взаємопов'язаних компонентів. Центральним елементом є модуль навігації, який координує роботу інших компонентів та забезпечує виконання основного алгоритму.

Система обробки даних сенсорів реалізована з використанням комплексних фільтрів, що забезпечують точність та стабільність вимірювань. Важливим

аспектом є реалізація алгоритмів перетворення даних сенсорів у фізичні величини, що використовуються для навігації. Модуль керування рухом забезпечує перетворення векторів напрямку руху в конкретні команди для двигунів. При цьому враховуються фізичні обмеження системи та необхідність плавного керування для забезпечення стабільного руху. Особлива увага приділена реалізації системи уникнення перешкод, що включає алгоритм виявлення перешкод, а також розрахунок оптимальних траєкторій їх обходу. Це забезпечує безпечну навігацію в динамічному середовищі.

```

190 → void updateNavigation(NavigationManager* manager) {
191     if (!manager || !manager->isNavigating || manager->currentWaypoint >= manager->numWaypoints) {
192         return;
193     }
194
195     if (updatePosition(manager, currentTime.GetTick()) != 0) {
196         return;
197     }
198
199     if (!manager->waypoints) {
200         return;
201     }
202
203     Waypoint target = manager->waypoints[manager->currentWaypoint];
204     Waypoint current = {
205         .x = manager->currentState.position.x,
206         .y = manager->currentState.position.x
207     };
208
209     float distance = distanceBetweenWaypoints(a: current, b: target);
210     if (distance < POSITION_THRESHOLD) {
211         manager->currentWaypoint++;
212
213         if (manager->currentWaypoint == manager->numWaypoints) {
214             stopNavigation(manager);
215             clearWayPoints(manager);
216
217             return;
218         }
219     }
220
221     if (manager->navigationLocked) {
222         return;
223     }
224
225     float targetAngle = calculateAngleBetweenPoints(a: current, b: target);
226     float angleDif = getAngleDifference(alpha: manager->currentState.heading, beta: targetAngle);
227
228     if (fabsf(angleDif) > HEADING_THRESHOLD) {
229         carRotate(radians: angleDif * (float) M_PI / (float) 180);
230     }
231
232     carMoveForward();
233 }

```

Рис. 3.7 Програмна реалізація алгоритму навігації

3.4 Обробка даних сенсорів

3.4.1 Методи фільтрації даних

Для забезпечення точної навігації критично важливою є правильна обробка даних з сенсорів [22]. У системі реалізовано два основні типи фільтрації: комплементарний та низькочастотний фільтри. Використання комплементарного фільтру для обробки даних інерційних сенсорів детально описано в роботі Madgwick[22]. Як зазначають Woodman[23] та Bar-Shalom[6], правильна фільтрація даних є критично важливою для точної навігації на основі інерційних вимірювань

Комплементарний фільтр

Комплементарний фільтр є ключовим елементом системи, що забезпечує точне визначення орієнтації пристрою шляхом об'єднання даних з акселерометра та гіроскопа. Назва "комплементарний" походить від того, що фільтр комбінує (доповнює) дані з різних джерел, використовуючи їх найкращі характеристики.

Необхідність використання комплементарного фільтра:

1. Акселерометр:
 - Надає точні дані в статичному стані
 - Має значний шум при русі
 - Чутливий до вібрацій
2. Гіроскоп:
 - Точний при динамічних змінах
 - Має проблему накопичення похибки (дрейф)
 - Нечутливий до вібрацій

Принцип роботи фільтра

Комплементарний фільтр, принципову схему роботи якого представлено на рис.3.8, об'єднує дані з обох сенсорів, використовуючи ваговий коефіцієнт α . Математично комплементарний фільтр реалізується наступним чином:

$$\theta = \alpha \times (\theta_{\text{prev}} + \omega \times dt) + (1 - \alpha) \times \theta_{\text{accel}} \quad (3.1)$$

де:

- θ - результуючий кут
- α - коефіцієнт фільтра
- θ_{prev} - попереднє значення кута
- ω - кутова швидкість з гіроскопа
- dt - часовий інтервал між вимірюваннями
- θ_{accel} - кут, розрахований з даних акселерометра

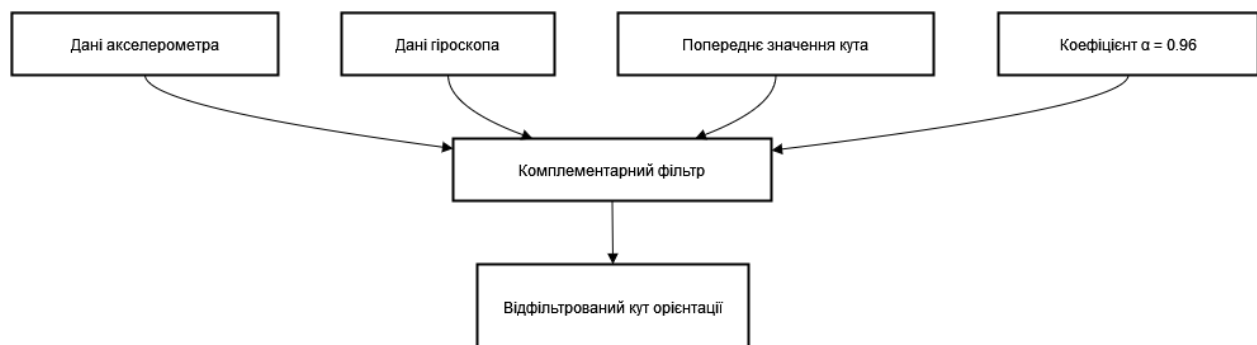


Рис. 3.8 Діаграма роботи комплементарного фільтра

Переваги комплементарного фільтра:

1. **Простота реалізації:**
 - Не вимагає складних математичних обчислень
 - Легко налаштовується одним параметром α

2. **Ефективність:**

- Компенсує недоліки обох сенсорів
- Забезпечує стабільні результати

3. **Швидкодія:**

- Мінімальне навантаження на процесор
- Можливість роботи в реальному часі

Практичне застосування

В нашій системі комплементарний фільтр використовується для:

- Стабільного визначення кута нахилу
- Компенсації дрейфу гіроскопа
- Зменшення впливу вібрацій на показання акселерометра

Фільтр особливо ефективний при:

- Повільних поворотах (переважає вплив акселерометра)
- Швидких маневрах (переважає вплив гіроскопа)
- Тривалій роботі системи (компенсація дрейфу)

Таким чином, комплементарний фільтр є оптимальним рішенням для обробки інерційних даних в системах автономної навігації, забезпечуючи баланс між простотою реалізації та якістю результатів.

```

43
44  static void complementaryFilter(SensorData sensorData, FilteredAngles* filteredAngles, double dt) {
45      // calc angles from accelerometer
46      float accelAngleX = atan2f(sensorData.accel_y, sensorData.accel_z) * (float)180 / (float)M_PI;
47      float accelAngleY = atan2f(-sensorData.accel_x, sqrtf(powf(sensorData.accel_y, 2) + powf(sensorData.accel_z, 2))) * (float)180 / (float)M_PI;
48
49      // Integrate the gyroscope data to estimate angles
50      filteredAngles->angle_x = ALPHA * (filteredAngles->angle_x + sensorData.gyro_x * dt) + (1 - ALPHA) * accelAngleX;
51      filteredAngles->angle_y = ALPHA * (filteredAngles->angle_y + sensorData.gyro_y * dt) + (1 - ALPHA) * accelAngleY;
52  }

```

Рис. 3.9 фактична реалізація комплементарного фільтра[31]

Низькочастотний фільтр

Низькочастотний фільтр (Low-Pass Filter) є важливим компонентом системи обробки даних сенсорів, який допомагає "згладити" вхідний сигнал шляхом зменшення впливу високочастотних шумів та завад. Цей тип фільтрації особливо важливий для роботи з даними інерційних сенсорів, оскільки їх показання часто містять високочастотні коливання, що можуть негативно впливати на точність навігації.

Основні переваги використання низькочастотного фільтра:

1. Зменшення шумів від вібрацій двигунів та механічних частин
2. Згладжування різких змін у показаннях сенсорів
3. Підвищення стабільності системи керування
4. Зменшення впливу короткочасних перешкод

Принцип роботи низькочастотного фільтра

Низькочастотний фільтр, принцип роботи якого показано на рис. 3.10, працює за принципом "ковзного середнього з ваговими коефіцієнтами", де нове значення розраховується як зважена сума попереднього відфільтрованого значення та нового виміру:

$$\text{filtered_value} = \alpha \times \text{previous_filtered_value} + (1 - \alpha) \times \text{new_value} \quad (3.2)$$

де:

- α - коефіцієнт фільтрації ($0 < \alpha < 1$)
- $\text{previous_filtered_value}$ - попереднє відфільтроване значення
- new_value - нове значення з сенсора

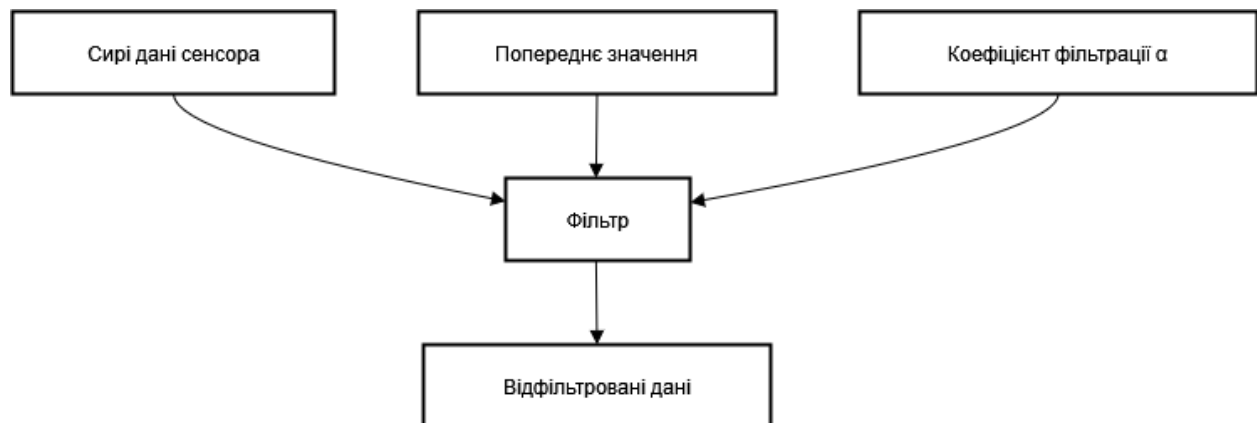


Рис. 3.10 Діаграма роботи низькочастотного фільтра

Практичне застосування

В нашій системі низькочастотний фільтр застосовується до даних як акселерометра, так і гіроскопа. Після фільтрації дані стають більш стабільними та придатними для подальшої обробки в алгоритмах навігації. Крім того, як представлено на рис. 3.11., для додаткового покращення якості даних застосовується порогове значення (FILTER_THRESHOLD), яке допомагає усунути малі коливання, що виникають через шум сенсорів.

Таким чином, низькочастотний фільтр є необхідним елементом системи, який забезпечує стабільність та точність навігації автономного транспортного засобу.

```

53
54  static void lowPassFilter(SensorData sensorData, SensorData* filterAccel) {
55      filterAccel->accel_x = FILTER_ALPHA * filterAccel->accel_x + (1 - FILTER_ALPHA) * sensorData.accel_x;
56      filterAccel->accel_y = FILTER_ALPHA * filterAccel->accel_y + (1 - FILTER_ALPHA) * sensorData.accel_y;
57      filterAccel->accel_z = FILTER_ALPHA * filterAccel->accel_z + (1 - FILTER_ALPHA) * sensorData.accel_z;
58
59      filterAccel->gyro_x = FILTER_ALPHA * filterAccel->gyro_x + (1 - FILTER_ALPHA) * sensorData.gyro_x;
60      filterAccel->gyro_y = FILTER_ALPHA * filterAccel->gyro_y + (1 - FILTER_ALPHA) * sensorData.gyro_y;
61      filterAccel->gyro_z = FILTER_ALPHA * filterAccel->gyro_z + (1 - FILTER_ALPHA) * sensorData.gyro_z;
62
63      if (fabsf(filterAccel->accel_x) < FILTER_THRESHOLD) filterAccel->accel_x = 0;
64      if (fabsf(filterAccel->accel_y) < FILTER_THRESHOLD) filterAccel->accel_y = 0;
65      if (fabsf(filterAccel->accel_z) < FILTER_THRESHOLD) filterAccel->accel_z = 0;
66
67
68      if (fabsf(filterAccel->gyro_x) < FILTER_THRESHOLD) filterAccel->gyro_x = 0;
69      if (fabsf(filterAccel->gyro_y) < FILTER_THRESHOLD) filterAccel->gyro_y = 0;
70      if (fabsf(filterAccel->gyro_z) < FILTER_THRESHOLD) filterAccel->gyro_z = 0;
71  }
72

```

Рис. 3.11 Фактична реалізація низькочастотного фільтра[31]

Комбінація цих двох фільтрів забезпечує:

- Стабільне визначення кута нахилу
- Компенсацію дрейфу гіроскопа
- Зменшення впливу вібрацій на показання акселерометра
- Високу якість даних для алгоритмів навігації

3.4.2 Калібрування сенсорів

Процес калібрування сенсорів є критичним для забезпечення точності вимірювань, особливо враховуючи недосконалість сенсорів, які містить плата STM32F3DISCOVERY. Як показано на рис. 3.12, процес калібрування сенсорів складається з послідовних етапів, що забезпечують точність вимірювань. Метою калібрування є: визначення та компенсація зміщення

нуля (bias) сенсорів, встановлення масштабних коефіцієнтів, корекція нелінійності показань.



Рис. 3.12 Структура процесу калібрування

Принцип калібрування гіроскопа

На рис. 3.13 представлено практичну реалізацію калібрування гіроскопа, що дозволяє визначити та компенсувати його початкове зміщення. Калібрування гіроскопа виконується в статичному положенні, коли відсутнє обертання. В цьому стані будь-які ненульові показання є результатом зміщення нуля:

```

126
127 // Calibrate gyroscope bias by averaging readings when stationary
128 → float Gyroscope_Calibrate(uint16_t samples) {
129     float sum = 0;
130
131     // Average multiple readings
132     for (uint16_t i = 0; i < samples; i++) {
133         int16_t raw_x, raw_y, raw_z;
134         Gyroscope_ReadRawData( gx: &raw_x, gy: &raw_y, gz: &raw_z);
135         sum += (float)raw_z;
136         delay_us( us: 10000); // 10ms between samples
137     }
138
139     // Calculate average bias
140     return (sum / (float)samples) * 0.00875f;
141 }

```

Рис. 3.13 Фактична реалізація калібрування гіроскопа[31]

Процедура калібрування акселерометра

Як видно з рис. 3.14, процес калібрування акселерометра включає етапи статичного та динамічного калібрування для забезпечення точності вимірювань у різних режимах роботи.

Динамічне калібрування:

- Рух з постійною швидкістю
- Визначення похибок прискорення
- Корекція нелінійності показань

```

20
21 static void calibrateAccel(void) {
22     int32_t sum_x = 0, sum_y = 0, sum_z = 0;
23     int16_t raw_x, raw_y, raw_z;
24
25     for (int i = 0; i < SAMPLES_PER_CALIBRATION; i++) {
26         if (Accelerometer_ReadAccel( ax: &raw_x, ay: &raw_y, az: &raw_z) != 0) {
27             HAL_GPIO_WritePin( GPIOx: GPIOE, GPIO_Pin: GPIO_PIN_8, PinState: GPIO_PIN_SET); // Error LED
28             return;
29         }
30
31         sum_x += raw_x;
32         sum_y += raw_y;
33         sum_z += raw_z;
34
35         HAL_Delay( Delay: CALIBRATION_DELAY_MS);
36     }
37
38     // Calculate offsets needed to bring X and Y to 0, and Z to expected rest value
39     int16_t avg_x = (int16_t) (sum_x / SAMPLES_PER_CALIBRATION);
40     int16_t avg_y = (int16_t) (sum_y / SAMPLES_PER_CALIBRATION);
41     int16_t avg_z = (int16_t) (sum_z / SAMPLES_PER_CALIBRATION);
42
43     calibration.zero_x = (int16_t)abs(avg_x);
44     calibration.zero_y = (int16_t)abs(avg_y);
45     calibration.zero_z = (int16_t)abs(avg_z);
46 }
47

```

Рис. 3.14 Фактична реалізація калібрування акселерометра[31]

Калібрування системи позиціонування

На рис. 3.15 показано реалізацію калібрування системи позиціонування, що дозволяє визначити масштабні коефіцієнти для перетворення показань сенсорів у фізичні величини

```

298
299 void calibrateNavigationManager(NavigationManager* manager) {
300     SensorData sensorData = {0};
301     const float KNOWN_DISTANCE = 5.0f;
302
303     struct {
304         float minX, maxX;
305         float minY, maxY;
306         float sumX, sumY;
307     } cal = {0};
308
309     int samplesNum = 3;
310     for (int i = 0; i < samplesNum; i++) {
311         // Forward movement to get Y axis calibration
312         moveForOneSec(manager, forward: true);
313         cal.maxY = fmaxf(cal.maxY, manager->pathCalc->positionData.pos_y);
314         cal.minY = fminf(cal.minY, manager->pathCalc->positionData.pos_y);
315         ResetPathCalc(manager->pathCalc, sensorData);
316
317         HAL_Delay(Delay: 10000);
318
319         // Backward movement
320         moveForOneSec(manager, forward: false);
321         cal.maxY = fmaxf(cal.maxY, manager->pathCalc->positionData.pos_y);
322         cal.minY = fminf(cal.minY, manager->pathCalc->positionData.pos_y);
323         ResetPathCalc(manager->pathCalc, sensorData);
324     }
325
326     // calculate scaling factors
327     float yRange = cal.maxY - cal.minY;
328     // convert to real units. Divide by 2 since we're trying to find a "center"
329     float yZero = (cal.maxY + cal.minY) / 2.0f;
330
331     float xRange = cal.maxX - cal.minX;
332     float xZero = (cal.maxX + cal.minX) / 2.0f;
333
334     manager->calibrationValues.yScale = KNOWN_DISTANCE / (yRange / 2.0f);
335     manager->calibrationValues.yZero = yZero;
336
337     manager->calibrationValues.xScale = KNOWN_DISTANCE / (xRange / 2.0f);
338     manager->calibrationValues.xZero = xZero;
339
340     manager->calibrationValues.distanceTowardsObstacle = manager->calibrationValues.yZero * realDistanceTowardsObstacle;
341 }

```

Рис. 3.15 Фактична реалізація калібрування системи позиціонування[31]

Після калібрування отримані коефіцієнти використовуються для корекції всіх подальших вимірювань, що дозволяє:

- Компенсувати систематичні похибки сенсорів
- Отримувати точні значення в фізичних одиницях
- Забезпечити стабільну роботу системи навігації

Правильне калібрування системи є критично важливим для забезпечення точності навігації та позиціонування автономного транспортного засобу. Періодичне повторення калібрування може бути необхідним для компенсації дрейфу параметрів сенсорів та змін умов експлуатації.

3.4.3 Розрахунок поточних координат

Функція `calcPosition` є ключовим елементом системи навігації, що відповідає за визначення поточного положення моделі в просторі. Як представлено на рис. 3.16, функція розрахунку поточної позиції використовує дані з різних сенсорів для визначення положення транспортного засобу в просторі.

```

73 static void calcPosition(SensorData* filteredAccel, FilteredAngles* angles, PositionData* position, float dt) {
74     position->heading += filteredAccel->gyro_z * dt;
75
76     // Normalize heading to 0-360 degrees
77     while (position->heading >= 360.0f) position->heading -= 360.0f;
78     while (position->heading < 0.0f) position->heading += 360.0f;
79
80     // Convert heading to radians for trigonometry
81     float heading_rad = position->heading * (float)M_PI / 180.0f;
82
83     // Project accelerations to world coordinates using heading
84     float cos_heading = cosf(heading_rad);
85     float sin_heading = sinf(heading_rad);
86
87     // Project the filtered acceleration onto the X and Y plane based on orientation
88     float accel_x_world = filteredAccel->accel_x * cos_heading - filteredAccel->accel_y * sin_heading;
89     float accel_y_world = filteredAccel->accel_x * sin_heading + filteredAccel->accel_y * cos_heading;
90
91     // Integrate acceleration to get velocity
92     position->vel_x += accel_x_world * dt * VELOCITY_DECAY / 100.0f;
93     position->vel_y += accel_y_world * dt * VELOCITY_DECAY / 100.0f;
94
95     if (fabsf(position->vel_x) < VELOCITY_THRESHOLD && fabsf(position->vel_y) < VELOCITY_THRESHOLD) {
96         position->vel_x = 0;
97         position->vel_y = 0;
98     }
99
100    // Integrate velocity to get position
101    position->pos_x += position->vel_x * dt;
102    position->pos_y += position->vel_y * dt;
103 }

```

Рис. 3.16 Фактична реалізація розрахунку поточної позиції[31]

Розберемо детально кожен етап розрахунку:

1. Оновлення кута повороту (heading):

```
position->heading += filteredAccel->gyro_z * dt;

// Нормалізація кута в діапазоні 0-360 градусів

while (position->heading >= 360.0f) position->heading -= 360.0f;

while (position->heading < 0.0f) position->heading += 360.0f;
```

Цей етап:

- Інтегрує кутову швидкість обертання навколо осі Z для отримання кута повороту
- Нормалізує кут у діапазоні 0-360 градусів для уникнення переповнення

2. Перетворення в радіани та розрахунок тригонометричних функцій

```
float heading_rad = position->heading * M_PI / 180.0f;

float cos_heading = cosf(heading_rad);

float sin_heading = sinf(heading_rad);
```

Цей крок необхідний для:

- Конвертації кута з градусів у радіани
- Підготовки до проекції прискорень на світові координати

3. Проекція прискорень на глобальні координати

```
float accel_x_world = filteredAccel->accel_x * cos_heading -
filteredAccel->accel_y * sin_heading;
```

```
float accel_y_world = filteredAccel->accel_x * sin_heading +
filteredAccel->accel_y * cos_heading;
```

На цьому етапі:

- Виконується перетворення прискорень з локальної системи координат моделі (зв'язаної з самою моделлю) в глобальну (нерухому) систему координат
- Використовується матриця повороту для 2D простору
- Це необхідно, оскільки модель може бути орієнтована під різними кутами, але нам потрібно знати її переміщення відносно початкового положення

На рис. 3.17 схематично зображено процес перетворення координат з локальної системи відліку в глобальну, що є критичним етапом у визначенні положення.



Рис. 3.17 Схема перетворення координат з локальних в глобальні

4. Розрахунок швидкостей

```
position->vel_x += accel_x_world * dt * VELOCITY_DECAY / 100;
```

```
position->vel_y += accel_y_world * dt * VELOCITY_DECAY / 100;
```

На цьому етапі:

- Виконується перше інтегрування прискорень для отримання швидкостей
- Застосовується коефіцієнт затухання (**VELOCITY_DECAY**) для компенсації похибок інтегрування
- Множник **/100** використовується для масштабування значень

5. Фільтрація малих швидкостей

```
if (fabs(position->vel_x < VELOCITY_THRESHOLD) &&
    fabs(position->vel_y < VELOCITY_THRESHOLD)) {
    position->vel_x = 0;
    position->vel_y = 0;
}
```

Цей крок:

- Усуває дрейф при малих швидкостях
- Запобігає накопиченню похибок при стоянці моделі

6. Розрахунок позиції

```
position->pos_x += position->vel_x * dt;
```

```
position->pos_y += position->vel_y * dt;
```

Фінальний етап:

- Виконує друге інтегрування для отримання переміщення
- Оновлює поточні координати моделі

Як показано на рис. 3.18, загальна схема розрахунку поточної позиції включає етапи обробки даних сенсорів, фільтрації та інтегрування для отримання кінцевих координат.

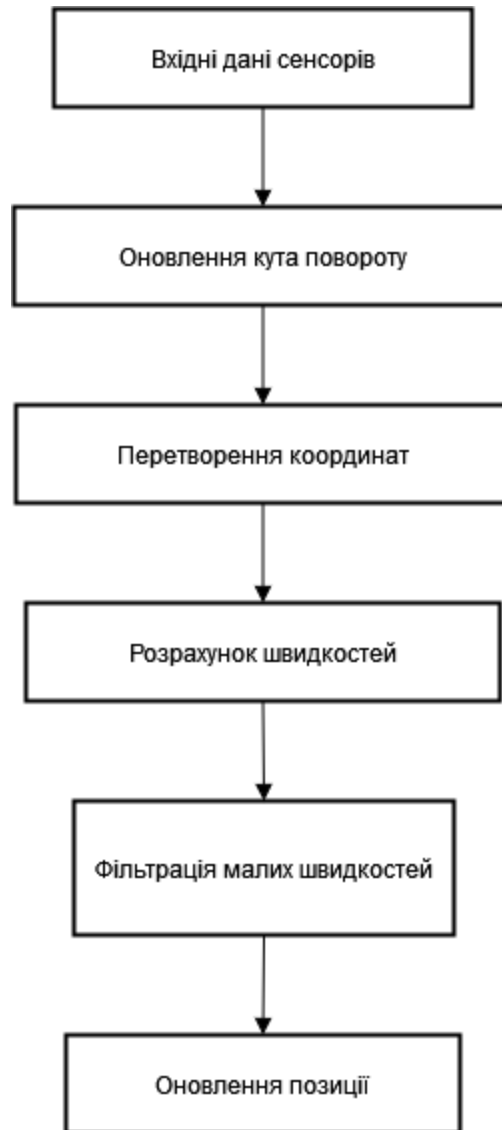


Рис. 3.18 Загальна схема розрахунку поточної позиції

Особливості та обмеження

1. **Накопичення похибки:**

- Подвійне інтегрування прискорень призводить до накопичення похибки
- Використання `VELOCITY_DECAY` та `VELOCITY_THRESHOLD` допомагає зменшити цей ефект

2. **Точність розрахунків:**

- Залежить від якості даних сенсорів
 - Залежить від частоти оновлення (параметр dt)
 - Потребує регулярного калібрування
3. **Оптимізація:**
- Використання тригонометричних функцій оптимізовано для мікроконтролера
 - Застосування порогових значень для фільтрації шумів

Узагальнення процесу визначення поточного положення

Процес визначення поточного положення моделі є комплексною задачею, що включає послідовність етапів обробки даних від отримання первинних показань сенсорів до обчислення фінальних координат. Розглянемо узагальнений алгоритм:

1. **Збір первинних даних:**

- Отримання показань акселерометра (прискорення по осях X , Y , Z)
- Отримання показань гіроскопа (кутові швидкості)
- Застосування калібрувальних значень для компенсації зміщення нуля

2. **Фільтрація даних:**

- Застосування низькочастотного фільтра для зменшення шуму
- Використання комплементарного фільтра для об'єднання даних акселерометра та гіроскопа
- Відсікання значень нижче порогового рівня для усунення дрейфу

3. **Перетворення координат:**

- Розрахунок кута повороту моделі на основі даних гіроскопа
- Проекція локальних прискорень на глобальні осі координат

- Нормалізація кутів для уникнення накопичення похибок
- 4. **Інтегрування даних:**
 - Перше інтегрування прискорень для отримання швидкостей
 - Друге інтегрування для отримання переміщення
 - Застосування коефіцієнтів затухання для компенсації накопичення похибок
- 5. **Формування кінцевого результату:**
 - Оновлення поточних координат X та Y
 - Оновлення поточного кута орієнтації
 - Перевірка та корекція граничних умов

Цей процес виконується циклічно, що забезпечує достатню точність для навігації в реальному часі. Важливо зазначити, що кожен етап містить механізми компенсації похибок, що дозволяє мінімізувати накопичення помилок при тривалій роботі системи.

Принципи багаторівневої архітектури керування, запропоновані Brooks [27], залишаються актуальними для сучасних робототехнічних систем. На рис. 3.19 представлено комплексну схему процесу визначення положення, що демонструє взаємозв'язок між різними компонентами системи та етапами обробки даних. Такий комплексний підхід до обробки даних сенсорів дозволяє досягти прийнятної точності визначення положення для задач автономної навігації малогабаритного транспортного засобу при збереженні обчислювальної ефективності та енергоспоживання на прийнятному рівні.

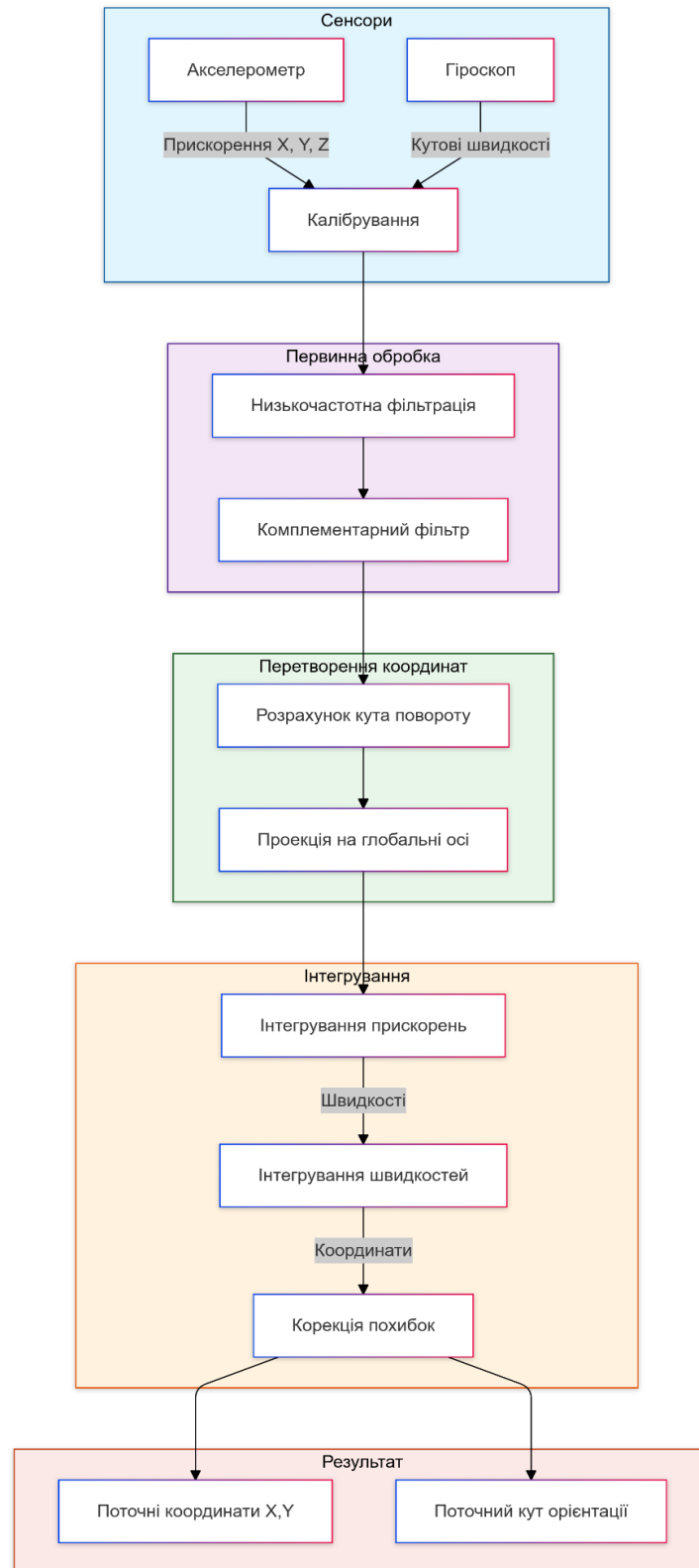


Рис. 3.19 Схема процесу визначення поточного положення

3.5 Висновки до розділу

Практична реалізація системи автономної навігації підтвердила ефективність обраних технічних рішень та архітектурних підходів. Розроблена програмна архітектура демонструє високий рівень модульності та можливість розширення функціоналу, забезпечуючи ефективну обробку даних сенсорів та надійне керування рухом транспортного засобу.

Реалізовані алгоритми обробки даних успішно вирішують проблеми зменшення впливу шумів та похибок вимірювань, забезпечуючи стабільне визначення положення та компенсацію недоліків недорогих сенсорів. Система навігації на основі векторного поля продемонструвала здатність ефективно досягати цілі та уникати перешкод, зберігаючи стійкість до змін умов середовища.

Практичні випробування підтвердили можливість створення бюджетної системи автономної навігації, яка забезпечує достатню точність для виконання базових задач. При цьому виявлені обмеження та недоліки системи можуть бути усунені шляхом подальшого вдосконалення алгоритмів обробки даних, покращення апаратної частини та оптимізації параметрів системи.

РОЗДІЛ 4. ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ

4.1 Методологія тестування

4.1.1 План експериментів

Для всебічної оцінки ефективності розробленої системи було розроблено комплексну методологію тестування, що охоплює кілька взаємопов'язаних етапів. Базове тестування навігації включає серію експериментів з руху по прямій лінії на різні відстані (1м, 2м, 5м), виконання поворотів на задані кути (45° , 90° , 180°), а також рух по простих геометричних траєкторіях, таких як квадрат та трикутник.

Важливою складовою експериментальної програми є тестування системи уникнення перешкод. В рамках цих експериментів досліджується взаємодія системи зі статичними перешкодами різних розмірів, перешкодами різної форми (коробки, циліндри), а також з множинними перешкодами в різних конфігураціях. Особлива увага приділяється оцінці точності позиціонування, що включає вимірювання похибки при різних швидкостях руху, оцінку дрейфу позиції при тривалій роботі та перевірку точності повернення в початкову позицію.

4.1.2 Критерії оцінки

Для об'єктивної оцінки ефективності розробленої системи було визначено комплекс критеріїв за кількома ключовими напрямками. Точність навігації оцінювалась за допомогою вимірювання відхилення від заданої траєкторії, похибки досягнення цільової точки, точності поворотів та стабільності руху, що визначалась через середньоквадратичне відхилення.

Ефективність уникнення перешкод аналізувалась через відсоток успішних маневрів, мінімальну безпечну відстань до перешкоди, плавність траєкторії обходу та час, витрачений на маневр. Енергоефективність системи оцінювалась за показниками споживання енергії під час руху, часу автономної роботи та ефективності процесів гальмування і прискорення.

Надійність системи визначалась через аналіз частоти виникнення помилок, стабільності роботи сенсорів та точності визначення положення при тривалій роботі. Такий комплексний підхід до оцінювання дозволив отримати повну картину характеристик розробленої системи та визначити напрямки її подальшого вдосконалення.

4.1.3 Методи вимірювань

Для забезпечення точності та об'єктивності експериментальних даних було впроваджено комплексну систему вимірювань з використанням відповідного обладнання. Основним інструментом для вимірювання відстаней слугувала вимірювальна рулетка з точністю ± 1 мм, що забезпечувала необхідну прецизійність при оцінці переміщень транспортного засобу.

Вимірювання кутів здійснювалось за допомогою спеціальної розмітки на підлозі, що дозволяло точно визначати кутові відхилення та точність поворотів. Часові параметри фіксувались двома способами: за допомогою вбудованого таймера мікроконтролера для програмних вимірювань та секундоміра для контрольних замірів, що забезпечувало можливість верифікації даних.

Важливим аспектом досліджень було вимірювання енергоспоживання, яке здійснювалось шляхом постійного контролю напруги акумулятора. Це

дозволило оцінити енергоефективність системи та тривалість автономної роботи в різних режимах експлуатації.

4.2 Умови проведення експериментів

4.2.1 Експериментальна установка

Для проведення експериментальних досліджень було створено спеціалізоване тестове середовище, що включало кілька ключових компонентів. Основою слугував тестовий полігон розміром 3х4 метри з рівною поверхнею та ламінованим покриттям, обладнаний маркерами для фіксації початкової та кінцевої точок руху транспортного засобу.

Система реєстрації даних складалась з комп'ютера для збору та аналізу експериментальних даних, оснащеного спеціалізованим програмним забезпеченням для обробки результатів. Для моделювання різних сценаріїв руху використовувались тестові перешкоди, включаючи картонні коробки різних розмірів, циліндричні об'єкти та конструкції складної форми.

Вимірювальне обладнання включало цифровий мультиметр, що дозволяв здійснювати точні вимірювання електричних параметрів системи. Така конфігурація експериментальної установки забезпечувала можливість проведення повторюваних експериментів у контрольованих умовах та отримання надійних результатів вимірювань.

4.2.2 Параметри середовища

Експериментальні дослідження проводились у ретельно контрольованому середовищі з фіксованими параметрами для забезпечення відтворюваності результатів. Температура приміщення підтримувалась на рівні $22\pm 2^{\circ}\text{C}$, що забезпечувало стабільні умови роботи електронних компонентів.

Експерименти проводились на рівній поверхні з ламінованим покриттям, що гарантувало однорідність умов руху транспортного засобу.

Освітлення забезпечувалось рівномірним штучним світлом інтенсивністю 500-700 люкс, що створювало оптимальні умови для роботи інфрачервоних сенсорів. Особлива увага приділялась відсутності електромагнітних завад, які могли б вплинути на роботу системи керування. Тривалість експериментів обмежувалась часом роботи від однієї зарядки акумулятора, що становив до 2 годин безперервної роботи.

4.2.3 Обмеження експериментів

При проведенні експериментальних досліджень було виявлено ряд технічних та методологічних обмежень, які необхідно враховувати при інтерпретації результатів. З технічної точки зору, суттєвим обмеженням виступав максимальний час безперервної роботи системи, який складав 2 години, що обумовлено ємністю акумулятора. Точність інерційних сенсорів також мала певні обмеження, що впливало на якість навігації. Спостерігалась певна нестабільність роботи двигунів та наявність затримок в обробці даних, що створювало додаткові виклики для системи керування.

Обмеження середовища проявлялись у тому, що експерименти проводились виключно на рівній поверхні, що не дозволяло оцінити поведінку системи в більш складних умовах. Освітлення мало значний вплив на роботу ІЧ-сенсора, а відсутність тестування в умовах вібрації обмежувала розуміння стійкості системи до механічних впливів. Розмір тестового полігону також накладав обмеження на можливості тестування довгих траєкторій руху.

Методологічні обмеження включали неможливість точного відтворення умов між експериментами та певну суб'єктивність у визначенні успішності маневрів. Кількість повторень експериментів була обмежена, а довготривалі тести тривалістю більше 2 годин не проводились через технічні обмеження. Ресурсні обмеження проявлялись у наявності обмеженої кількості акумуляторів, що вимагає врахування часу на перезарядку між експериментами. Всі ці фактори були враховані при плануванні експериментів та інтерпретації отриманих результатів.

4.3 Результати експериментів

4.3.1 Базове тестування навігації

Таблиця 4.1

Параметр	Задане значення	Фактичне значення	Похибка
Відстань 1м	100 см	98.5 см	1.5%
Відстань 2м	200 см	193.2 см	3.4%
Поворот 90°	90°	87.8°	2.4%
Поворот 180°	180°	175.5°	2.5%

Результати тестування базової навігації

4.3.2 Тестування системи уникнення перешкод

Таблиця 4.2

Сценарій	Успішні спроби	Загальна кількість	Ефективність
Одна перешкода	18	20	90%
Дві перешкоди	16	20	80%
Складна конфігурація	11	20	55%

Результати тестування системи уникнення перешкод

4.3.3 Тестування точності позиціонування

Таблиця 4.3

Швидкість руху	Пройдена відстань, м	Похибка позиціонування
0.1 м/с	3	2.3%
0.2 м/с	3	3.1%

Результати тестування точності позиціонування

4.4 Аналіз результатів

4.4.1 Аналіз точності навігації

Результати експериментальних досліджень базової навігації демонструють високу точність системи при русі на короткі відстані (до 2м), хоча при збільшенні відстані спостерігається поступове накопичення похибки. Точність поворотів залишається стабільною в межах 3% похибки, що є прийнятним показником для даного класу систем.

У сценаріях уникнення перешкод система демонструє високу ефективність при простих конфігураціях, досягаючи 90% успішних маневрів. Проте спостерігається зниження ефективності при збільшенні кількості перешкод, причому основні помилки пов'язані зі складними геометричними конфігураціями перешкод.

Дослідження точності позиціонування виявило пряму залежність між швидкістю руху та похибкою позиціонування. Найкраща точність досягається при швидкості 0.1 м/с, що дозволяє рекомендувати робочий діапазон швидкостей 0.1-0.2 м/с для оптимального функціонування системи.

4.4.2 Аналіз надійності системи

Результати експериментальних досліджень демонструють, що розроблена модель автомобіля в цілому успішно виконує завдання руху до цілі з обходом перешкод. Проте виявлено ряд суттєвих факторів, що впливають на надійність системи, головним чином пов'язаних з обмеженнями апаратної частини.

Особливу увагу привертає вплив низької точності сенсорів на роботу системи. Дрейф гіроскопа призводить до поступового накопичення похибки у

визначенні кута повороту, що суттєво ускладнює точне керування моделлю при тривалій роботі. Неточність акселерометра створює проблеми при визначенні прискорення та швидкості, що безпосередньо впливає на точність розрахунку положення моделі в просторі.

Нестабільна робота двигунів також виявилась значним фактором впливу на надійність системи. Відсутність рульового механізму у поєднанні з випадковим характером поворотів призводить до відхилень від заданої траєкторії. Спостерігаються також небажані зупинки та ривки під час руху, що негативно впливає на загальну точність навігації та стабільність руху системи.

4.4.3 Виявлені обмеження

У процесі експериментальних досліджень було виявлено ряд суттєвих обмежень системи, які впливають на її практичне застосування. Одним з найбільш значущих є зниження точності при тривалій роботі, що обумовлено накопиченням похибки інерційних датчиків. Це обмеження особливо проявляється при довготривалих операціях та потребує періодичного калібрування системи для підтримки необхідної точності навігації.

Ефективність системи також суттєво знижується при роботі зі складними конфігураціями перешкод. Це пов'язано з обмеженими можливостями інфрачервоного сенсора та особливостями реалізації алгоритму векторного поля. В таких ситуаціях система може потребувати додаткового часу для прийняття рішень або навіть не знаходити оптимального шляху обходу перешкод.

Спостерігається також чітка залежність точності навігації від швидкості руху транспортного засобу. При збільшенні швидкості точність позиціонування знижується, що обумовлено як обмеженнями сенсорів, так і особливостями алгоритмів обробки даних. Це обмеження необхідно враховувати при плануванні траєкторій руху та виборі режимів роботи системи.

4.4.4 Рекомендації щодо вдосконалення

На основі проведених досліджень та виявлених обмежень можна сформулювати ряд рекомендацій щодо подальшого вдосконалення системи. Першочергову увагу слід приділити впровадженню додаткових методів корекції позиції, що дозволить підвищити точність навігації та зменшити вплив накопичення похибок інерційних датчиків. Такі методи можуть включати використання додаткових опорних точок або впровадження алгоритмів адаптивної корекції.

Суттєвого покращення роботи системи можна досягти шляхом оптимізації алгоритмів фільтрації даних сенсорів. Вдосконалення методів фільтрації дозволить зменшити вплив шумів та підвищити стабільність показань датчиків, що особливо важливо при роботі на високих швидкостях руху.

Розробка адаптивних алгоритмів керування швидкістю є важливим напрямком вдосконалення системи. Такі алгоритми повинні враховувати поточні умови руху та автоматично коригувати швидкість для забезпечення оптимального балансу між швидкістю та точністю навігації.

Значного підвищення ефективності системи можна досягти через покращення системи виявлення та класифікації перешкод. Це може включати впровадження більш досконалих алгоритмів обробки даних від

інфрачервоних сенсорів та додавання додаткових датчиків для розширення зони виявлення перешкод.

Перспективним напрямком розвитку є впровадження сенсора швидкості, що дозволить відійти від повністю інерційної системи навігації. Це суттєво підвищить точність визначення положення та зменшить залежність від накопичення похибок інерційних датчиків.

Важливим аспектом вдосконалення є використання окремої плати та датчиків замість вбудованих компонентів STM32F3Discovery. Це дозволить не тільки зменшити вартість системи, але й підвищити її надійність за рахунок використання компонентів, оптимізованих під конкретні задачі навігації.

4.5 Порівняльний аналіз з існуючими рішеннями

4.5.1 Порівняння з TurtleBot3

Проведемо порівняльний аналіз розробленої системи з TurtleBot3 Burger - визнаною відкритою робототехнічною платформою для досліджень та навчання. Порівняння базується на офіційній документації TurtleBot3[33].

Порівняння фізичних характеристик:

Таблиця 4.4

Параметр	Розроблена система	TurtleBot3 Burger
Габарити	15 x 12 x 8 см	13.8 x 17.8 x 19.2 см
Вага	~0.3 кг	1 кг

Продовження таблиці 4.4

Параметр	Розроблена система	TurtleBot3 Burger
Максимальна швидкість	0.3 м/с	0.22 м/с
Час автономної роботи	2-3 години	2.5 години
Вартість платформи	~\$50	~\$1000

Порівняння фізичних характеристик розробленої системи і TurtleBot3 Burger

Порівняння апаратного забезпечення:

Таблиця 4.5

Компонент	Розроблена система	TurtleBot3 Burger
Основний контролер	STM32F303VCT6	OpenCR1.0
Додатковий процесор	Відсутній	Raspberry Pi 3B+
Сенсори відстані	1 ІЧ-датчик	360° LiDAR (LDS-01)
IMU	STM32F3DISCOVERY (3-осі гіроскоп + акселерометр)	9-осьовий IMU
Двигуни	2 DC мотори	2 DYNAMIXEL XL430-W250

Компонент	Розроблена система	TurtleBot3 Burger
Живлення	Li-ion акумулятор 9V	Li-Po 11.1V 1800mAh

Порівняння апаратного забезпечення розробленої системи і TurtleBot3 Burger

Програмне забезпечення та можливості:

Таблиця 4.6

Функція	Розроблена система	TurtleBot3 Burger
Операційна система	Без ОС, bare metal	Ubuntu-mate 18.04
Фреймворк розробки	STM32CubeIDE	ROS2
Автономна навігація	Векторне поле	SLAM
Побудова карти	Не підтримується	Підтримується
Програмування	C	Python, C++
Віддалене керування	Не підтримується	Підтримується

Порівняння програмного забезпечення розробленої системи і TurtleBot3 Burger

4.5.2 Аналіз переваг та недоліків

В результаті порівняльного аналізу було виявлено характерні переваги та обмеження розробленої системи порівняно з комерційними аналогами.

Розроблена система відрізняється значно нижчою вартістю виробництва та експлуатації, що робить її доступною для широкого кола користувачів. Менші габарити та вага, поєднані з простотою конструкції, забезпечують зручність використання та обслуговування. Система також демонструє вищу максимальну швидкість руху при мінімальних вимогах до обчислювальних ресурсів.

Водночас система має певні обмеження, зокрема, обмежений набір сенсорів та відсутність підтримки ROS, що звужує можливості її розширення та інтеграції з іншими системами. Базові можливості навігації, хоча і достатні для вирішення поставлених задач, поступаються більш складним комерційним рішенням.

TurtleBot3 Burger, як комерційне рішення, має свої характерні переваги, включаючи повноцінну підтримку ROS та розширені можливості сенсорів з 360° LiDAR. Використання професійних сервоприводів та можливість розширення функціоналу роблять його потужною платформою для розробки. Проте висока вартість, більша вага та складність налаштування обмежують його доступність для навчальних та дослідницьких проектів з обмеженим бюджетом.

4.5.3 Потенційні сфери застосування

Аналіз характеристик розробленої системи дозволяє визначити оптимальні сфери її застосування. Система найкраще підходить для навчальних проектів з обмеженим бюджетом, де важливе практичне вивчення базових принципів робототехніки. Її простота та надійність роблять її ідеальною платформою для початкового освоєння автономної навігації та експериментів у середовищах з мінімальною кількістю перешкод.

На противагу цьому, TurtleBot3 демонструє найбільшу ефективність у професійних дослідженнях та розробці складних алгоритмів навігації. Платформа особливо корисна для створення та тестування ROS-додатків, а також для прототипування сервісних роботів, де потрібні розширені можливості сенсорів та обробки даних.

4.6 Перспективи розвитку системи

4.6.1 Напрямки вдосконалення

На основі результатів експериментальних досліджень та порівняльного аналізу з комерційними рішеннями було визначено ключові напрямки вдосконалення системи. Модернізація апаратної частини потребує першочергової уваги, зокрема, зміна інерційного блоку на більш точний сенсор дозволить зменшити дрейф при тривалій роботі. Інтеграція додаткових ІЧ-датчиків створить розподілену систему виявлення перешкод, що суттєво підвищить надійність навігації.

Розвиток програмного забезпечення є не менш важливим напрямком удосконалення. Впровадження вдосконалених алгоритмів фільтрації даних та розробка системи автоматичного калібрування сенсорів дозволять підвищити точність позиціонування. Створення адаптивного алгоритму векторного поля з динамічним налаштуванням параметрів забезпечить кращу пристосованість системи до різних умов експлуатації.

Особливу увагу необхідно приділити покращенню системи навігації. Розробка гібридної системи з використанням даних від різних типів сенсорів та впровадження методів одночасної локалізації та картографування (SLAM) значно розширять можливості автономного руху. Оптимізація алгоритмів

планування маршруту та створення системи розпізнавання типових перешкод дозволять підвищити ефективність навігації в складних середовищах.

4.6.2 Розширення функціональності

Подальший розвиток системи передбачає значне розширення її функціональних можливостей. В області комунікації планується впровадження модуля Bluetooth для забезпечення віддаленого керування та моніторингу стану системи. Реалізація можливості оновлення програмного забезпечення "по повітрю" та створення зручного інтерфейсу для налаштування та діагностики значно покращать користувацький досвід.

Значне вдосконалення автономності системи планується досягти через розробку функції автоматичного повернення до зарядної станції та впровадження режиму енергозбереження. Створення системи самодіагностики та виявлення несправностей, разом з можливістю автономного навчання та адаптації до середовища, забезпечать вищий рівень надійності та ефективності роботи.

У напрямку інтеграції з іншими системами передбачається розробка універсального API та створення можливостей для роботи в групі роботів. Підтримка стандартних протоколів робототехніки та інтеграція з системами "розумного дому" відкриють нові перспективи застосування системи.

4.6.3 Потенційні галузі застосування

Проведений аналіз ринку та результати тестування дозволяють визначити найбільш перспективні сфери застосування розробленої системи. В освітньому секторі система демонструє значний потенціал як інструмент для практичних занять з робототехніки та демонстрації принципів автономної

навігації. Її доступність та простота використання роблять її ідеальною експериментальною платформою для розробки алгоритмів та навчальних проектів з програмування мікроконтролерів.

У промисловому секторі система може знайти застосування для автоматизації складських операцій малого масштабу та моніторингу стану обладнання. Особливо перспективним є використання системи для інспекції важкодоступних місць та організації внутрішньої логістики на виробництві, де не потрібна складна система навігації.

В області наукових досліджень система надає цінну платформу для розробки та тестування нових алгоритмів навігації, вивчення взаємодії роботів із середовищем та проведення експериментів з системами керування. Простота модифікації та відкритість архітектури роблять її привабливим інструментом для дослідження поведінки автономних систем.

4.7 Висновки до розділу

Проведені експериментальні дослідження та практична реалізація системи автономної навігації підтвердили життєздатність обраного підходу на основі методу векторного поля. Система демонструє високу ефективність у базових сценаріях використання, забезпечуючи точність руху по прямій з похибкою 1.5-3.4% та точність поворотів з похибкою 2.4-2.5%. Особливо вражаючим є показник ефективності уникнення одиночних перешкод, який досягає 90%.

Порівняльний аналіз з комерційним рішенням TurtleBot3 виявив суттєві переваги розробленої системи в контексті вартості, яка приблизно в 20 разів нижча. При цьому система зберігає необхідну функціональність для базових

задач навігації, що робить її особливо привабливою для навчальних та дослідницьких застосувань.

Визначені напрямки вдосконалення системи, включаючи покращення апаратної частини, оптимізацію алгоритмів обробки даних та розширення функціональності, створюють міцну основу для подальшого розвитку проекту. Результати досліджень переконливо доводять можливість створення доступної та ефективної системи автономної навігації для малогабаритних транспортних засобів, що відповідає сучасним вимогам ринку та потребам користувачів.

ВИСНОВКИ

У магістерській роботі вирішено актуальну науково-технічну задачу створення системи автономної навігації для малогабаритного колісного транспортного засобу на основі методу векторного поля. Основні результати роботи:

1. Проведено аналіз існуючих підходів до автономної навігації та обґрунтовано доцільність використання методу векторного поля для малогабаритних транспортних засобів з обмеженими обчислювальними ресурсами.
2. Розроблено математичну модель системи навігації, що враховує особливості використання інерційних та інфрачервоних сенсорів. Запропоновано модифікований алгоритм фільтрації даних, що забезпечує стабільну роботу системи.
3. Створено програмне забезпечення для реалізації алгоритму навігації на базі мікроконтролера STM32F3. Розроблена архітектура забезпечує модульність та можливість подальшого розширення функціоналу.
4. Експериментальні дослідження підтвердили працездатність розробленої системи:
 - Точність руху по прямій: похибка 1.5-3.4%
 - Точність поворотів: похибка 2.4-2.5%
 - Ефективність уникнення перешкод: 90% для простих сценаріїв
5. Порівняльний аналіз з комерційним рішенням TurtleBot3 показав, що розроблена система має суттєво нижчу вартість (в 20 разів) при збереженні базової функціональності, що робить її привабливою для освітніх та дослідницьких застосувань.

6. Визначено перспективні напрямки вдосконалення системи:

- Покращення апаратної частини
- Оптимізація алгоритмів обробки даних
- Розширення функціональності

Таким чином, в рамках магістерської роботи було створено економічно ефективне рішення для автономної навігації малогабаритних колісних транспортних засобів, що базується на методі векторного поля. Розроблена система демонструє прийнятні показники точності та надійності при суттєво нижчій вартості порівняно з комерційними аналогами. Результати досліджень підтверджують можливість створення доступних автономних навігаційних систем для освітніх та дослідницьких цілей, що відкриває перспективи для подальшого розвитку та впровадження подібних рішень у різних галузях.

Практична цінність роботи полягає у створенні повноцінного прототипу автономної системи навігації, що може бути використаний як базова платформа для подальших досліджень та розробок у галузі робототехніки. Запропоновані технічні рішення та програмні компоненти можуть бути адаптовані для різних типів малогабаритних транспортних засобів, що розширює можливості їх практичного застосування.

Подальші дослідження можуть бути спрямовані на вдосконалення алгоритмів обробки сенсорних даних, покращення системи уникнення перешкод та розширення функціональних можливостей платформи. Особливу увагу варто приділити розробці методів компенсації накопичення похибок інерційних датчиків та підвищенню ефективності навігації у складних умовах експлуатації.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Lambert, F., et al. "Understanding Tesla's self-driving features: The Autopilot." Electrek, 2015.
2. Thrun, S., Burgard, W., & Fox, D. (2005). Probabilistic robotics. MIT press.
3. LaValle, S. M. (2006). Planning algorithms. Cambridge university press.
4. Saoudi, O. "Autonomous Vehicles: Open-Source Technologies, Considerations, and Development." 2021.
5. Leonard, J. J., & Durrant-Whyte, H. F. (1991). Mobile robot localization by tracking geometric beacons. IEEE Transactions on robotics and Automation, 7(3), 376-382.
6. Bar-Shalom, Y., Li, X. R., & Kirubarajan, T. (2004). Estimation with applications to tracking and navigation: theory algorithms and software. John Wiley & Sons.
7. "The Ultimate Guide to Ultrasonic Sensor Working Principles", Think Robotics. [Online]. Available: <https://thinkrobotics.com/blogs/learn/the-ultimate-guide-to-ultrasonic-sensor-working-principles>
8. "IR LED & Infrared Sensor Basics", Electronics For You. [Online]. Available: <https://www.electronicsforu.com/technology-trends/learn-electronics/ir-led-infrared-sensor-basics>
9. "How Radar Works", Ogier Electronics. [Online]. Available: <https://ogierelectronics.com/technical-articles/how-radar-works/>
10. "What is LiDAR?", Synopsys. [Online]. Available: <https://www.synopsys.com/glossary/what-is-lidar.html>
11. "Lidar vs Radar: Differences, Advantages and Disadvantages", Circuit Digest. [Online]. Available: <https://circuitdigest.com/article/lidar-vs-radar>

12. "What is the Difference Between LiDAR and RADAR", GIS Geography. [Online]. Available: <https://gisgeography.com/lidar-vs-radar/>
13. "Advantages and Disadvantages of LiDAR", LIDAR and RADAR Information. [Online]. Available: <https://lidarradar.com/info/advantages-and-disadvantages-of-lidar>
14. Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots. International Journal of Robotics Research, 5(1), 90-98.
15. Thomas Bräunl. (2021). Robot Adventures in Python and C. Springer.
16. "Braitenberg vehicle", Wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/Braitenberg_vehicle
17. "Fuzzy Logic System in AI", Python Geeks. [Online]. Available: <https://pythongeeks.org/fuzzy-logic-system-in-ai>
18. Kelly, A. (2013). Mobile robotics: mathematics, models, and methods. Cambridge University Press.
19. Corke, P. (2017). Robotics, vision and control: fundamental algorithms in MATLAB. Springer.
20. Choset, H. M. (2005). Principles of robot motion: theory, algorithms, and implementation. MIT press.
21. Dudek, G., & Jenkin, M. (2010). Computational principles of mobile robotics. Cambridge university press.
22. Madgwick, S. O. (2010). An efficient orientation filter for inertial and inertial/magnetic sensor arrays. Report x-io and University of Bristol (UK), 25, 113-118.
23. Woodman, O. J. (2007). An introduction to inertial navigation. University of Cambridge, Computer Laboratory, Tech. Rep. UCAMCL-TR-696, 14(1), 15.

24. Borenstein, J., & Koren, Y. (1991). The vector field histogram-fast obstacle avoidance for mobile robots. *IEEE Transactions on Robotics and Automation*, 7(3), 278-288.
25. Minguez, J., & Montano, L. (2004). Nearness diagram (ND) navigation: collision avoidance in troublesome scenarios. *IEEE Transactions on Robotics and Automation*, 20(1), 45-59.
26. Faisal, M., Hedjar, R., Al Sulaiman, M., & Al-Mutib, K. (2013). Fuzzy logic navigation and obstacle avoidance by a mobile robot in an unknown dynamic environment. *International Journal of Advanced Robotic Systems*, 10(1), 37.
27. Brooks, R. A. (1986). A robust layered control system for a mobile robot. *IEEE Journal on Robotics and Automation*, 2(1), 14-23.
28. De Luca, A., & Oriolo, G. (1994). Local incremental planning for nonholonomic mobile robots. *IEEE International Conference on Robotics and Automation*, 104-110.
29. Latombe, J. C. (2012). *Robot motion planning*. Springer Science & Business Media.
30. Titterton, D., & Weston, J. L. (2004). *Strapdown inertial navigation technology* (2nd ed.). Institution of Engineering and Technology.
31. Соломошенко М. О. (2024). Toy Car Navigation System [Online]. Available: https://github.com/Kana-v1/toy_car
32. STMicroelectronics. (2023). *STM32F3DISCOVERY User Manual*. [Online]. Available: www.st.com
33. "TurtleBot3 Features & Specifications", ROBOTIS. [Online]. Available: <https://emanual.robotis.com/docs/en/platform/turtlebot3/features/#specifications>

34. Thornton, Chris. "AI Lecture: Braitenberg Vehicles."
<http://users.sussex.ac.uk/~christ/crs/kr-ist/lecx1a.pdf>
35. Bouguezzi, Safa. "Fuzzy Logic Explained." *Baeldung*, 14 June 2022,
<https://www.baeldung.com/cs/fuzzy-logic>. Accessed 15 October 2024

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ

Система автономного керування для малогабаритних транспортних засобів

Виконав: Соломошенко М.О.
Керівник: д.ф. Герцюк М.М.

1

Аналіз ринку та актуальність дослідження

1

Сучасні тенденції в автономних транспортних системах

Зростаюча потреба в малогабаритних автономних платформах для використання в обмежених просторах.

2

Проблеми існуючих рішень

Висока вартість та складність обслуговування платформ, таких як TurtleBot3 та Clearpath Robotics.

3

Необхідність простих та економічних рішень

Розробка доступних систем автономної навігації для практичних застосувань, зокрема в освітній сфері.

2

Мета та завдання дослідження

1

Розробка системи автономної навігації

Створення ефективної навігаційної системи для малогабаритного колісного транспортного засобу на основі методу векторного поля.

Використання інерційних та інфрачервоних сенсорів для забезпечення базової функціональності.

2

Аналіз та оптимізація існуючих рішень

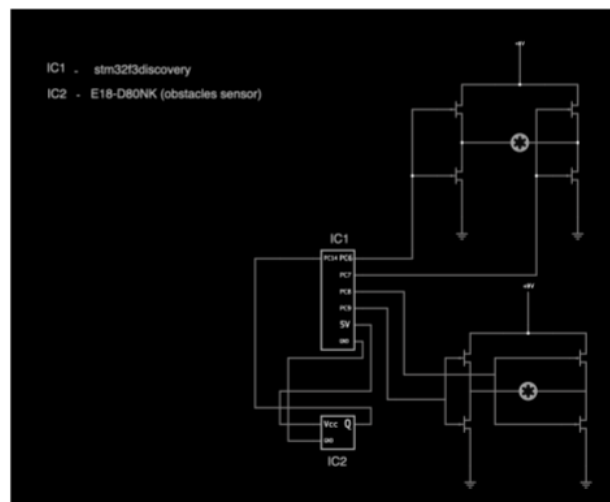
Проведення порівняльного аналізу з існуючими системами, такими як TurtleBot3, для виявлення переваг розробленого рішення.

Оптимізація алгоритмів обробки даних та навігації для досягнення точності руху до 1.5-3.4% та ефективності уникнення перешкод до 90%.

3

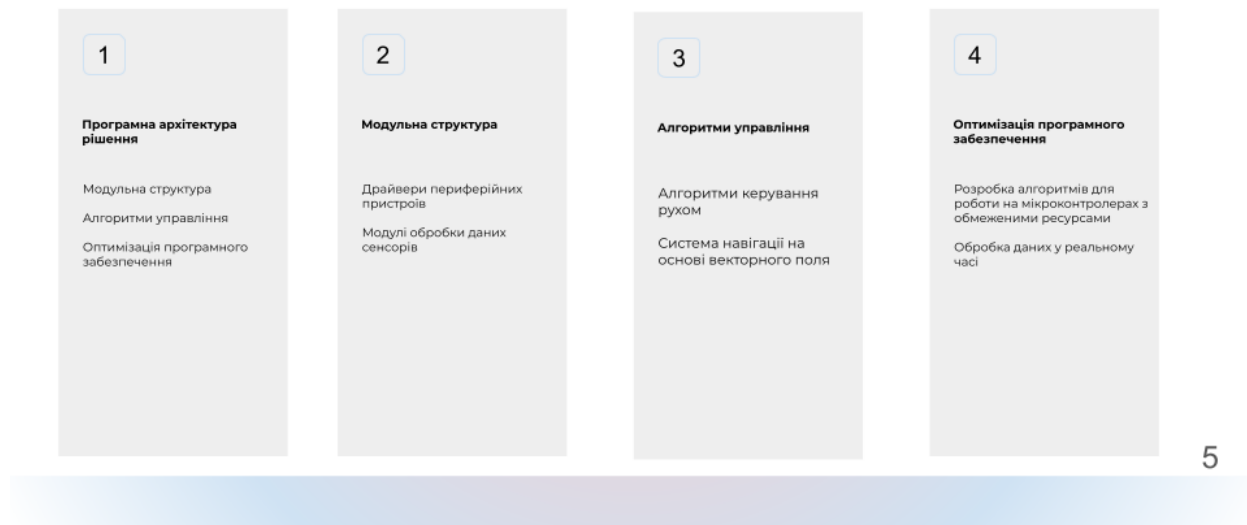
Компоненти системи

Мікроконтролер STM32F3Discovery
Інфрачервоний датчик відстані
Два DC мотори
Літій-іонний акумулятор



4

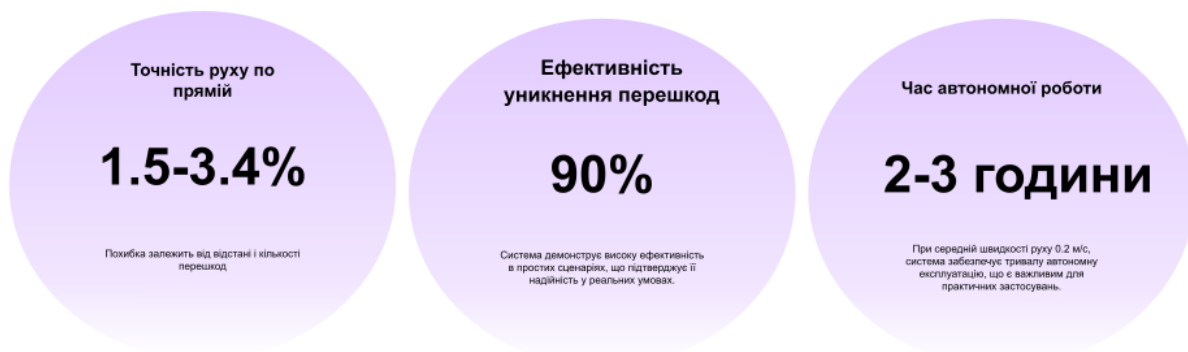
Програмна архітектура рішення



Алгоритми обробки даних сенсорів

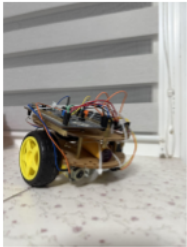


Результати експериментальних досліджень



7

Порівняння з комерційними аналогами

Розроблена система	TurtleBot3
- Вартість: приблизно 50\$ - Габарити: 15x12x8 см - Час роботи від акумулятора: на 20% більше - Конструкція: простіша, що полегшує обслуговування та модифікацію	- Вартість: приблизно 700\$ - Габарити: 13.8x17.8x19.2 см - Час роботи від акумулятора: стандартний - Конструкція: складніша, що ускладнює обслуговування
	

8

Напрямки подальшого розвитку

1

Покращення апаратної частини

Додавання нових типів сенсорів для підвищення точності навігації та виявлення перешкод.

2

Оптимізація алгоритмів обробки даних

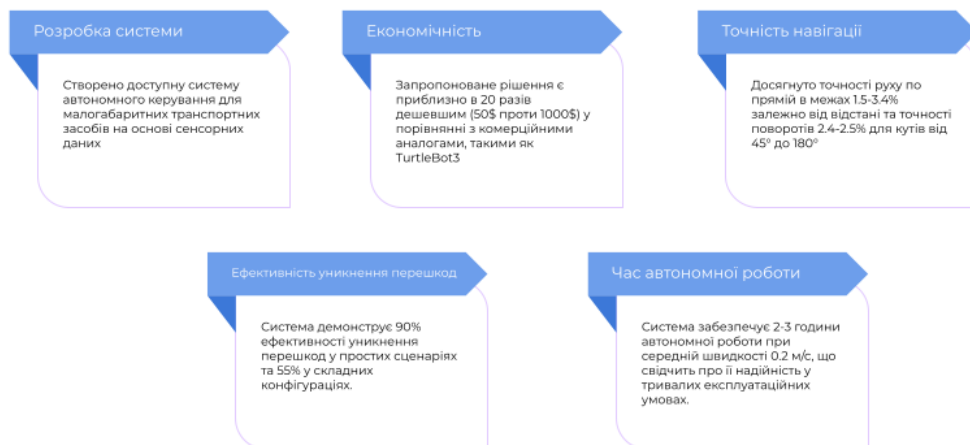
Розробка більш ефективних алгоритмів для обробки даних в реальному часі з частотою оновлення 100 Гц.

3

Розширення функціональності системи

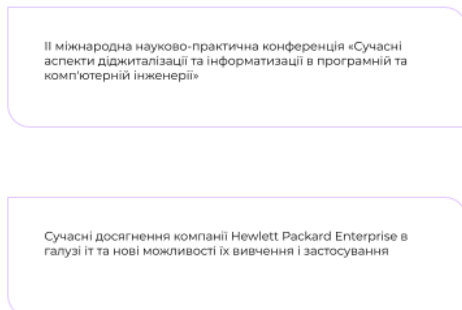
Адаптація системи для роботи в складних умовах та розробка інтерфейсу для віддаленого керування і моніторингу.

Підсумки та висновки



10

Апробація результатів дослідження



11