

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____ комп'ютерної інженерії та управління
(повна назва)

Кафедра _____ електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

Рівень вищої освіти _____ другий (магістерський)

Методи управління мобільним роботом в умовах
динамічних перешкод
(тема)

Виконав:
студент _____ ІІ _____ курсу, групи _____ СПм-23-2
Сорока К.Ю.
(прізвище, ініціали)

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

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

Освітня програма _____
Системне програмування
(повна назва освітньої програми)

Керівник: _____ проф. Каргін А.О.
(посада, прізвище, ініціали)

Допускається до захисту

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

2025 р.

Харківський національний університет радіоелектроніки

Факультет _____ комп'ютерної інженерії та управління _____

Кафедра _____ електронних обчислювальних машин _____

Рівень вищої освіти _____ другий (магістерський) _____

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

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

Освітня програма _____ Системне програмування _____
(повна назва)

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

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студенту _____ Сороці Катерині Юріївні _____
(прізвище, ім'я, по батькові)

1. Тема роботи _____ Методи управління мобільним роботом в умовах динамічних перешкод _____

затверджена наказом по університету від “ 22 ” листопада 2024 р. № 1236 Ст

2. Термін подання студентом роботи до екзаменаційної комісії _____ 20 січня 2025 р.

3. Вхідні дані до роботи _____

- 1. загальні поняття, що пов'язані з безпілотними літальними апаратам та динамічними умовами: РЕБ, БПЛА, глобальна навігаційна супутникова система, адаптивна перешкода;
- 2. технологія: dotNet;
- 3. фреймворк: Unity;
- 4. алгоритми: SLAM, A*, 3D-VFH.
- 5. мова програмування: C#

4. Перелік питань, що потрібно опрацювати у роботі _____

- 1. аналіз методів управління мобільних роботів та огляд засобів протидії;
- 2. огляд інструментів та аналіз існуючих рішень;
- 3. дослідження роботи програмного забезпечення та проведення експериментів;
- 4. висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп’ютерних ілюстрацій (слайдів) слайд-презентація – 12 слайдів.

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

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

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз проблеми та огляд існуючих рішень	26.11.2024-30.11.2024	
2	Аналіз завдання та пошук літератури	01.12.2024-04.12.2024	
3	Аналіз технічних засобів для реалізації	05.12.2024-07.12.2024	
4	Розробка програмних модулів	08.12.2024-19.12.2024	
5	Відлагодження програмних модулів	19.12.2024-28.12.2024	
6	Опрацювання результатів дослідження	29.12.2024-04.01.2025	
7	Оформлення матеріалів кваліфікаційної роботи	05.01.2024-20.01.2025	

Дата видачі завдання 25 листопада 2024 р.

Студент _____
(підпис)

Керівник роботи _____
(підпис)

проф. Каргін А.О.
(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 70 с., 10 рис., 2 табл., 2 дод., 15 джерел.

ДИНАМІЧНІ УМОВИ, 3D-VFH, АЛГОРИТМ, БЕЗПЛОТНИЙ ЛІТАЛЬНИЙ АПАРАТ, ЙМОВІРНІСНЕ ПЛАНУВАННЯ, ТРАЄКТОРІЯ.

Метою даної роботи є аналіз, розробка та оптимізація методів керування мобільним роботом у складних динамічних умовах. У дослідженні розглядаються недоліки та переваги як класичних алгоритмів управління, так і прогресивних підходів, включно з методами штучного інтелекту, адаптивного планування траєкторій.

У ході виконання кваліфікаційної роботи буде досліджено динамічні перешкоди, а також шляхи їх уникнення. Як стратегію подолання динамічних перешкод буде розглянуто алгоритм 3D-VFH. Та представлено власну програмну модифікацію та статистику подолання смуги бар'єрів. За критерії оптимального шляху буде приділено увагу мінімізації зіткнень, оптимізації, враховуючи критерії перешкод.

Доцільність дослідження полягає у тому, що при постійному зростанні складності задач, що стоять перед автономними БПЛА, як наприклад удосконалення засобів радіоелектронної боротьби чи постійне застосування кібернетичних атак, необхідно надати технологічну відповідь. Ці передумови формують потребу у вдосконаленні методів управління, які б забезпечували автономність, точність та оперативність роботи технічних систем.

ABSTRACT

Master's thesis: 70 pages, 10 figures, 2 tables, 2 appendices, 15 sources.

DYNAMIC CONDITIONS, 3D-VFH, ALGORITHM, UNMANNED AERIAL VEHICLE, PROBABILISTIC PLANNING, TRAJECTORY.

The objective of this research is to analyze, develop, and optimize methodologies for the control of a mobile robot in complex dynamic environments. The study will consider the advantages and disadvantages of classical control algorithms and advanced approaches, including artificial intelligence, adaptive trajectory planning.

The research will include the investigation of dynamic obstacles and the development of strategies for their avoidance, with a focus on the 3D-VFH algorithm as a potential solution. The study will also present a modified program and statistical data on obstacle course navigation. In the context of optimal path planning, the research will prioritize minimizing collisions and optimization, with the consideration of obstacle criteria.

The study's significance stems from the mounting complexity of tasks confronting autonomous UAVs, such as the advancement of electronic warfare and the persistent threat of cyberattacks. This necessitates a technological response, which in turn demands the enhancement of control methodologies to ensure the autonomy, precision, and efficacy of technical systems.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	8
ВСТУП	9
1 КОНЦЕПЦІЯ АВТОНОМНОЇ НАВІГАЦІЇ	10
1.1 Класифікація БПЛА	10
1.2 Протидія безпілотним літальним апаратам	12
1.3 Кібернетичні атаки	14
1.4 Фізичне ураження	17
1.5 Засоби електронного захисту БЛПА	19
2 АРХІТЕКТУРНІ РІШЕННЯ БПЛА	21
2.1 Концепція автономної навігації	21
2.2 Перспективи автономної навігації та штучний інтелект	24
2.3 Існуючі реалізації	26
3 АЛГОРИТМИ АВТОНОМНОЇ НАВІГАЦІЇ	28
3.1 Проблематика та актуальність у сфері безпілотних літальних апаратів	28
3.2 Алгоритм SLAM	29
3.3 Візуальна одометрія	30
3.4 Алгоритми планування траєкторії	31
3.4.1 Класичний метод. A*	33
3.4.2. Статистичний метод Rapidly-Exploring Random Trees	34
4 РОЗРОБКА АЛГОРИТМУ	35
4.1 Виклики забезпечення надійності навігаційних систем	35
4.2 Алгоритм 3D-VFH	34
4.3 Середовище моделювання та технічні характеристики	38
4.4 Модифікація алгоритму 3D-VFH для покращення уникнення перешкод	40

4.5 Вибір цілі та математична постановка задачі	42
4.6 Аналіз якості алгоритму	46
4.7 Порівняння з іншими алгоритмами.....	52
ВИСНОВКИ.....	55
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	57
ДОДАТОК А ГРАФІЧНИЙ МАТЕРІАЛ КВАЛІФІКАЦІЙНОЇ РОБОТИ	59
ДОДАТОК Б DRONE DISCRETE SEARCH	66
Б.1 Фрагменти коду створеного програмного застосунку.....	66
ДОДАТОК В АПРОБАЦІЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕНЬ В ПУБЛІКАЦІЇ.....	70

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

АПР – адаптивна радарна перешкода

БПЛА – безпілотний літальний апарат

РЕБ – радіоелектронна боротьба

ППО – протиповітряна оборона

ІІ – штучний інтелект

AUDS – радарний комплекс (англ., Anti-Unmanned Aerial Vehicle Defense System)

CV – комп’ютерне бачення (англ., Computer Vision)

GPS – система глобального позиціювання (англ., Global Positioning System)

GNSS – супутникова система навігації (англ., Global Navigation Satellite System)

INS – інерціальна навігаційна система (англ., Inertial Navigation Systems)

FPGA – програмована користувачем вентильна матриця (англ., Field-Programmable Gate Array)

RL – навчання з підкріпленням (англ., Reinforcement learning)

RRT – (англ., Rapidly-Exploring Random Tree)

VBN – (англ., Vision-Based Navigation)

VDM – (англ., Vehicle Dynamic Model)

ВСТУП

Мобільна робототехніка в умовах сьогодення відіграє ключову роль у багатьох галузях, таких як промисловість, логістика, сільське господарство, охорона здоров'я та військова справа. Саме в останній сфері застосування радіокерованих моделей набуло особливо важливого значення для безпеки нашої держави.

Дослідники, котрі працюють над забезпеченням ефективної роботи мобільних роботів, у процесі проектування та тестування стикаються з питанням гарантування успішної навігації у складному динамічному середовищі, яке включає як рухомі, так і статичні перешкоди. Динамічні перешкоди бувають як інтерактивними та можуть змінювати свою поведінку залежно від зовнішніх чинників та дій об'єктів.

Актуальність теми кваліфікаційної роботи зумовлена щораз вищими вимогами до автономності мобільних роботів у різних середовищах, зокрема в контексті змінного оточення. Наприклад, у сфері автоматизованого доставлення вантажів безпілотною має демонструвати здатність оперативно адаптувати маршрут у випадку появи несподіваних перешкод. Аналогічно, у контексті рятувальних операцій автономні платформи повинні враховувати непередбачувані динамічні фактори для забезпечення ефективного виконання поставлених завдань.

Об'єктом дослідження є процес керування мобільним роботом у середовищі з динамічними перешкодами.

Предметом дослідження є методи управління, які дозволяють забезпечити ефективну, безпечну навігацію БПЛА в таких обставинах.

Результати даного дослідження можуть бути використані для вдосконалення наявних робото технічних систем, що застосовуються у складних і динамічних середовищах, сприяючи подальшому розвитку автономних технологій.

1 КОНЦЕПЦІЯ АВТОНОМНОЇ НАВІГАЦІЇ

1.1 Класифікація БПЛА

Безпілотний літальний апарат — це авіаційний засіб, здатний здійснювати зліт, політ і посадку без безпосередньої присутності оператора на борту. Широкі можливості безпілотного літального апарату часто порушують низку важливих аспектів питань. Одним із найбільш суперечливих тезисів є застосування БПЛА у військових операціях, що викликає занепокоєння щодо можливих порушень міжнародного гуманітарного права, зокрема через ризик помилкової ідентифікації цілей або завдання шкоди цивільному населенню. Крім того, відсутність безпосереднього ризику для операторів може стимулювати частіше застосування сили, що підвищує ймовірність ескалації конфліктів. Окрему увагу привертає питання відповідальності за помилки, які можуть виникати під час використання дронів: залишається неясним, хто саме відповідає за такі дії — оператор, розробник технології чи військове командування. Використання автономних систем ускладнює це питання, адже у випадку, коли рішення приймає штучний інтелект, відповідальність стає ще важче визначити.

Значна кількість факторів, таких як: нестабільна ситуація у світі та терористичні загрози — є головними рушіями технологічного розвитку у сфері безпілотних літальних апаратів різного призначення. Серед безпілотників виділяють: бойові, розвідувальні, транспортні [1]. В умовах бойових дій та заходів протидії тероризму особливе значення має забезпечення якісного та безпечного зв'язку. На тлі швидкого технічного прогресу та існуючих технологій відкривається значний потенціал для досліджень. Насамперед варто звернутися до офіційної воєнної документації міжнародної організації НАТО. Згідно зі стандартами STANAG 4670, БПЛА

класифікуються на:

Таблиця 1.1 – Класифікація БПЛА за STANAG 4670

Клас	Категорія	Рівень воєнних дій	Висота застосування	Радіус дії	Приклад платформ
Клас III (>600 кг)	ударні	стратегічний	До 20 км	необмежений	Reaper
	HALE	стратегічний	До 20 км	необмежений	Global Hawk
	MALE	оперативний	До 14 км	необмежений	Bayraktar TB2
Клас II (150-600 кг)	тактичний	тактичний	До 5.5 км	До 200 км	Hermes 450
Клас I (<150 кг)	Малі (>15 кг)	Тактичне формування	До 1.5 км	До 50 км	Scan Eagle DJI FlyCart 30
	Міні (<15 кг)	Тактичний підрозділ	До 900 м	До 25 км	Skylark
	Мікро (< 2 кг)	Тактичний підрозділ	До 60 м	До 5 км	Black Window DJI Mavic 3 Pro

У роботі буде розглянуто та опрацьовано методи управління мобільних роботів класу I та класу II – тактичного рівня воєнних дій. Доцільність дослідження мобільних роботів даних класів зумовлює революційність, яку здійснили компактні БПЛА за останні роки. Також у подальшій роботі буде приділено увагу і цивільним літальним апаратам, які також відносяться до класу I. Адже дані дрони мають відносно невелику вартість, гнучку

програмну конфігурацію та мульти-функціональність, а отже використовуються як для розвідки, так і транспортно-бойових цілей.

Усі сучасні моделі безпілотних літальних апаратів як правило обладнані GNSS системою. На основі сигналів від супутників, безпілотники обчислюють своє точне положення, висоту, швидкість і маршрут. За допомогою отриманої інформації, безпілотники обчислюють актуальні траєкторію польоту. Дана система має багато недоліків: пілотування таких БЛПА майже не можливе у міській місцевості з високими спорудами, з гірським ландшафтом, густою рослинністю – дані умови сприяють похибкам сигналу. Також супутниковий сигнал легко скомпрометувати, що знижує надійність даної навігаційної системи.

Питання нестабільності GNSS системи породжує низку рішень з використанням комп'ютерного зору, ШІ технологій, акселерометрів, гіроскопів, тощо. Такі рішення часто об'єднують у інерційні навігаційні системи – це автономні навігаційні системи, які не покладаються на зовнішню інформацію та не випромінюють енергію назовні [2].

1.2 Протидія безпілотним літальним апаратам

Для дослідження методів управління БЛПА в динамічних умовах, важливим завданням є аналіз засобів протидії дронам. Розгляд підходів оборони дозволяє висвітлити їхні переваги та недоліки, а отже допоможе у подальшій оптимізації навігаційної системи та структурі управління БЛПА. В залежності від виду та призначення безпілотників, використовуються різні методи оборони.

На сьогодні радіоелектронна боротьба у воєнній справі має найбільший потенціал для захисту від ворожих безпілотників. Системи РЕБ мають здатність:

- блокування сигналу управління між оператором і БЛПА, що призводить до втрати контролю над апаратом;

- глушіння GNSS-сигналів, що дезорієнтує апарат і унеможливорює точну навігацію;
- підміна сигналу, що змушує БЛПА рухатися у неправильно обраному напрямку або повернутися до точки зльоту.

Такі методи є ефективними проти більшості цивільних та комерційних БЛПА, оскільки ті мають обмежені можливості щодо захисту від втручання у сигнали. Серед сучасних систем РЕБ є низка показових прикладів. Дрон-рушниці, наприклад DroneGun Tactical від DroneShield, що здатні створювати перешкоди для частот, що використовуються літальними апаратами та блокують сигнали управління та навігації, спрямовуючи радіоперешкоди до цілі.

Система AUDS сумісного виробництва Великобританії та США, поєднує в собі радіолокаційне виявлення цілей, електрооптичне відстеження та здатність спрямованого радіочастотного придушення.



Рисунок 1.1 – Система радіоелектронної боротьби AUDS

Система AUDS, що представлена на рисунку 1.1 є прикладом сучасної інтеграції засобів боротьби з безпілотними літальними апаратами, що враховує еволюцію загроз у даній сфері. Розроблена для протидії дронам класів I та II, система демонструє здатність до адаптації та вдосконалення,

що робить її актуальною навіть для швидших і складніших цілей, таких як БПЛА класу III.

AUDS ефективно поєднує засоби електронної боротьби з кінетичними елементами, створюючи універсальну платформу для знешкодження дронів. Важливим аспектом є модульна архітектура системи, яка дозволяє встановлювати її на різні типи платформ, від мобільних одиниць, таких як STRYKER, до стаціонарних застосувань. Це надає військовим і урядовим установам гнучкість у розгортанні системи залежно від конкретних потреб.

Суттєвою перевагою AUDS є її здатність виявляти контролери дронів, глушити сигнали управління та знешкоджувати навіть автономні апарати, які використовують маршрутну навігацію або стільниковий зв'язок. У цій системі інтегровано три напрямки електронної боротьби: протидія саморобним вибуховим пристроям, боротьба з БПЛА та перешкоджання комунікаційним мережам противника. Така багатофункціональність знижує потребу в координації окремих засобів, дозволяючи командирам концентрувати ресурси на виконанні завдань – повідомляє Міністерство армії США на своєму офіційному сайті [3].

Таким чином, AUDS не лише пропонує ефективне рішення для сучасних загроз, але й закладає фундамент для протидії майбутнім викликам, пов'язаним із автономізацією БПЛА. Її спроможність об'єднувати летальні й нелетальні засоби робить систему ключовим елементом у розвитку комплексної оборони проти дронів.

1.3 Кібернетичні атаки

Із поширенням використання БПЛА виникає потреба в організації та реалізації методів їх кібербезпеки. Вразливості систем управління та зв'язку є можливостями для здійснення зловмисниками кібератак. Зазвичай це приводить до втрати контролю над пристроєм, витоку даних чи знищення апарату. Тож виникає потреба у аналізі найпоширеніших методів

кібернетичних атак на БПЛА та способів їм протидіяти.

В сучасних БПЛА найчастіше використовується радіочастотний зв'язок для керування та передачі чи отримання від пристрою даних. Найбільш поширеними типами атак на таким чином керовані системи найбільш є глушіння та перехоплення сигналу:

Під час глушіння зв'язку зловмисник генерує радіоперешкоди, що є потужніші за сигнал передбачуваного контролю. Така атака призводить до втрати зв'язку між оператором та безпілотником. Результат кібератаки зазвичай є аварія чи аварійна посадка.

Коли зловмисник намагається виконати перехоплення сигналу за допомогою спеціалізованих приймачів, він може отримати доступ до команд керування чи до телеметричних даних, що дозволяє цій особі вкрати дані чи заволодіти керуванням летальним апаратом.

Протидіяти таким атакам можна використовуючи захищені канали зв'язку з шифруванням та частотною модуляцією.

Орієнтація у просторі для більшості БПЛА може бути забезпечена завдяки глобальній навігаційній супутниковій системі (GNSS). Існують основні типи кібератак на подібні навігаційні системи – глушіння сигналів GNSS та їх підміна.

При такій атаці зловмисник відсилає підроблені сигнали навігації на пристрій. Вони імітують дані з довірених супутників. Результатом атаки часто є зміна траєкторії польоту або екстрена посадка БПЛА на землю, зазвичай в небезпечній точці.

Якщо зловмисник намагається здійснити перешкоду GNSS, він може спричинити створення радіоперешкод, наприклад у частоті 2.4 чи 5.8 ГГц. Вони роблять неможливими сприйняття сигналу з супутників і призводять до збою у навігації.

Захистом від подібних видів атак можуть виступати багатоканальні радіоприймачі, інерціальні навігаційні системи та програмні алгоритми, що виявляють підміну сигналів.

Зазвичай, більшість сучасних БПЛА мають на своєму бортовому комп'ютері вбудоване програмне забезпечення, що дозволяє автономне керування польотом та виконання різноманітних завдань. Наявність такого програмного забезпечення створює потенціал для знаходження та використання зломисниками його вразливостей. Серед найбільш поширених видів таких атак виділяють:

- зараження шкідливим програмним забезпеченням: зломисник завантажує шкідливе програмне забезпечення коли оновлюється система БПЛА чи коли отримує несанкціонований доступ до комп'ютера оператора. Найчастіше шкідливий код модифікує параметри польоту та викликає небажану поведінку апарата;

- використання вразливостей систем командного управління дроном: більшість БПЛА використовують стандартні протоколи зв'язку, що є уразливими до атак по типу "людина посередині" (Man-in-the-Middle). Зломисник перехоплює та змінює команди, які надсилаються БПЛА з пункту управління;

- атаки на пункт управління: через недостатню автентифікацію та захист комп'ютерів, атакуючі можуть отримати безпосередній доступ до системи керування безпілотником. Зломисники можуть примусово змінювати маршрути летальних апаратів чи зупиняти виконання поточних місій. Даний тип успішно виконаної атаки дає повний контроль над пристроєм.

Захищатися від загроз цього типу можна використовуючи багаторівневе шифрування зв'язку, автентифікацію користувачів і пристроїв, систематичне оновлення системних контрактів безпілотника з використанням цифрових підписів та розмежування доступу для критично важливих функцій системи.

Вбудовані в БПЛА пристрої, що використовуються для збору інформації про навколишнє середовище також є вразливим місцем сучасних безпілотників. Серед найпоширеніших атак виділяють атаки на камери, під

час яких використовують лазери для засліплення оптики або подачі помилкових зображень та маніпуляція з даними з сенсорів, коли зловмисники чинять вплив на дані за допомогою електромагнітних полів.

Таким чином кібернетичні загрози для автономних роботів містять досить широкий спектр методів: від простих фізичних перешкод до складних атак на систему пристрою. Щоб забезпечити надійний захист, необхідно впроваджувати комплексний підхід.

1.4 Фізичне ураження

У разі якщо попередні методи не були ефективними, можливе застосування методів фізичного ураження БЛПА, до яких відносять:

Зенітні комплекси – традиційний підхід для знищення великих БЛПА, який використовується військовими.

Високочастотні лазери – новітні технології, які можуть використовуватися для ураження менших або швидких БЛПА. Часто лазери будують за принципом «сітки», аби збільшити площину захисту.

Системи ППО з високочастотними випромінювачами – працюють за принципом руйнування електроніки на борту БЛПА, що робить його непридатним для виконання команд пілота.

Отже, фізичне ураження вважається ефективним засобом нейтралізації БЛПА, однак із недоліків можна виділити велику місткість ресурсів, що не завжди є виправданим проти безпілотних літальних апаратів класу I.

Різноманітність методів – від радіоелектронної боротьби до кібернетичних атак і фізичного знищення – дозволяє адаптувати протидію залежно від типу та призначення БЛПА. Водночас швидкий розвиток дронів та систем їхнього захисту вимагає постійного вдосконалення засобів протидії. «Гонитва» між РЕБ та БЛПА забезпечують постійну розробку та вдосконалення технологій, де кінцева мета – контроль повітряного простору.

Водночас швидкий розвиток безпілотників та систем їхнього захисту

вимагає постійного вдосконалення засобів протидії. Протистояння БПЛА та РЕБ є прикладом постійного розвитку технологій, які впливають одна на одну в умовах сучасного поля бою. У даному розділі буде опрацьовано основні етапи взаємодії БПЛА та РЕБ (на прикладі моделі РЕБ Terrascan):

Етап перший – виявлення БПЛА. РЕБ виявляє безпілота в зоні захисту, системи використовують радары та оптико-електронні засоби для визначення розташування, траєкторії та висоти польоту дрона. Розширені системи РЕБ можуть розрізняти типи дронів, що дозволяють обрати оптимальну тактику для їх нейтралізації. У той же час оператори дронів при досягненні зони впливу РЕБ зменшують електромагнітні сигнатури шляхом частотної адаптації та перемикання каналів, вмикають механізми захисту від перехоплення сигналу та GPS, задля уникнення виявлення.

Етап другий – аналіз сигналів та ідентифікація. Системи РЕБ перехоплюють та аналізують радіосигнали безпілота, ідентифікуючи їх джерело управління, частоту сигналу та вид навігації. Це дозволяє РЕБ не лише виявити апарат, а й визначити способи для порушення його управління або навігації. Сучасні дрони, у свою чергу, можуть мати кілька резервних каналів зв'язку та можливість перемикання між ними, що ускладнює завдання РЕБ.

Етап третій – порушення зв'язку та навігації. Після ідентифікації безпілота РЕБ переходить до фази активного придушення його сигналів управління та навігації. Для цього використовуються спрямовані антени, що глушать сигнали управління, частіше за все GNSS-сигнали. У випадку втрати сигналу навігації більшість апаратів автоматично повертаються до точки запуску або приземляються, що є стратегічною перевагою для РЕБ. Проте сучасні БПЛА, оснащені інерційними системами навігації, здатні продовжувати політ навіть при втраті GNSS-сигналів, що ускладнює завдання РЕБ.

Етап четвертий – підробка сигналів. Альтернативою глушіння сигналів, РЕБ може застосовувати підробку сигналів навігації, що спрямовує дрон на

хибний курс. Це може призвести до того, що дрон переміститься в іншу зону, вийде з зони бою або навіть приземлиться у визначеній оператором РЕБ точці. Для протидії підробки сигналів, сучасні дрони можуть використовувати багатоканальні GPS-антени, що виявляють несанкціоноване втручання і коригують траєкторію польоту.

Етап п'ятий – фізичне знешкодження. Коли методи радіопридушення та заміни сигналу неефективні, РЕБ може вдаватися до фізичного знешкодження БЛПА та використати зброю спрямованої енергії. До такої належать лазери, механічні засоби, та навіть сітки, випущені з наземних чи повітряних установок. Безпілотники можуть бути забезпечені системами уникнення перешкод та маневрувати, задля уникнення атаки. Проте ефективність даних систем може бути обмеженою при застосуванні високошвидкісних засобів ураження, що залишає дрони вразливими до ураження цими засобами.

Отже, після розгляду даних етапів, стає очевидним той факт, що протистояння між БЛПА і РЕБ тільки набирає складності, оскільки обидві технології швидко адаптуються та розвиваються. Це змагання технологій продовжує підштовхувати розробку все більш вдосконалених безпілотників та систем радіоелектронної боротьби, що збільшує ефективність захисту і наступальних операцій на сучасному полі бою.

1.5 Засоби електронного захисту БЛПА

Розглянемо засоби електронного захисту – де застосовуються методи та технології, що захищають дружні системи, наприклад БЛПА, від атак РЕБ з боку противника. Електронний захист спрямований на забезпечення надійного використання спектру дружніми силами, навіть в умовах інтенсивного придушення або перешкод противника.

Прикладом є високочастотна бойова мережева радіостанція, яка використовується збройними силами США та союзників SINCGARS, яку

використовують Збройні сили США, представлена на рисунку 1.2. Система радіозв'язку здійснює частотне перестрибування для забезпечення безпеки зв'язку навіть у ворожому середовищі.



Рисунок 1.2 – Високочастотна бойова мережева радіостанція SINCGARS

Також цікавим прикладом є технічний засіб, що уміщає у собі функції БЛПА, що оснащується РЕБ-навантаженням. Прикладом є модель MQ-9 Reaper. Таким чином даний безпілотною стає майже ідеальним апаратом розвідки та радіо-придушення по бою.

2 АРХІТЕКТУРНІ РІШЕННЯ БПЛА

2.1 Концепція автономної навігації

Сучасні безпілотні летальні апарати в нормальних умовах, зазвичай, керуються людиною-оператором. Однак в реальному світі часто трапляються ситуації, коли унеможлиблюється дистанційне керування роботом в ручному режимі через погодні умови, поганий зв'язок чи особливості місцевості. Тому виникає потреба впроваджувати бортові системи, що можуть керувати дроном та виконувати поставлені завдання без участі оператора.

Як правило, автономно керовані дрони оснащені датчиками щоб збирати інформацію та сприймати навколишнє середовище, дозволяючи їм самостійно орієнтуватися. Дані із датчиків направляються до бортового комп'ютера робота, далі на їх основі програмно приймається рішення яке визначає напрямок руху БПЛА, тягу рушійної установки тощо для виконання поставленого завдання.

Для аналізу технічних компонентів систем автономної навігації важливо детально розглянути архітектуру безпілотного літального апарата. Вибір архітектурного рішення суттєво впливає на можливості та ефективність системи, оскільки кожен компонент має виконувати свою функцію з урахуванням динаміки польоту, середовища експлуатації та поставлених завдань.

Зокрема, архітектура, зображена на рисунку 2.1, демонструє загальний підхід до організації компонентів БПЛА і представлена в дослідженні з ознайомчою метою. Ця схема є спрощеним відображенням проекту UTM (Unmanned Aircraft System Traffic Management) від NASA і дозволяє отримати базове уявлення про структуру системи безпілота [4].

Однак, на практиці архітектурні рішення можуть суттєво відрізнятися залежно від призначення БПЛА, необхідно дотримати баланс між

складністю, вартістю та ефективністю системи, а її вибір повинен ґрунтуватися на чіткому розумінні задач, які має виконувати апарат.

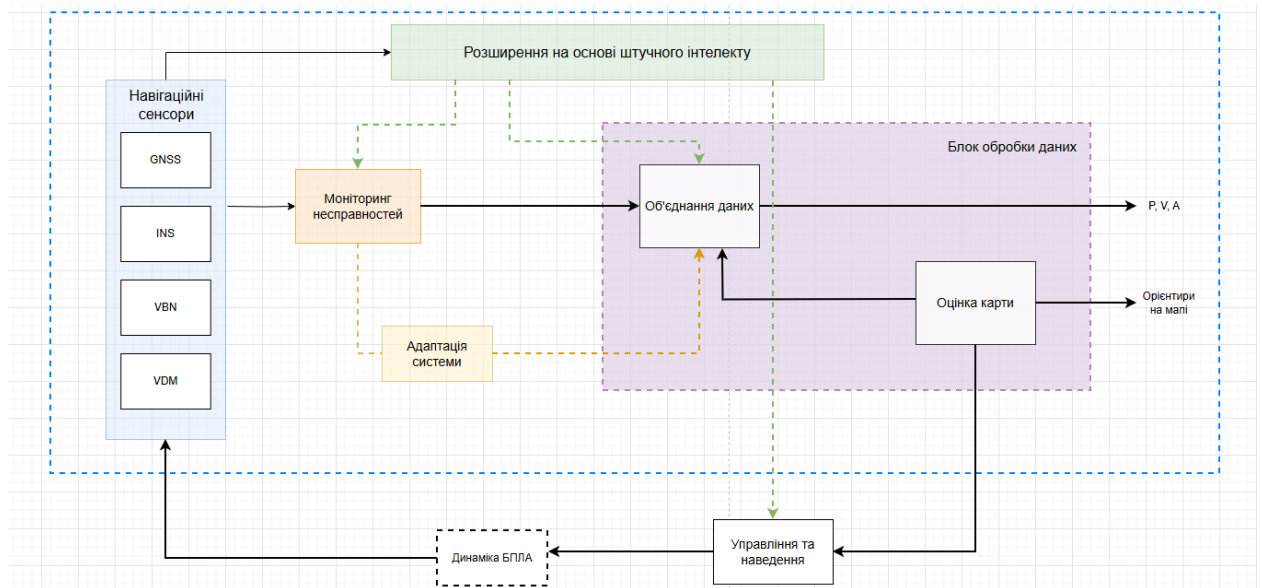


Рисунок 2.1 – Загальна архітектура безпілотного літального апарату

Отже, на схематичному зображенні можна виділити чотири категорії сенсорів та підсистем, що зазвичай використовуються для оцінки місцезнаходження та аналізу поточного середовища:

- GNSS (Global Navigation Satellite System) – це система супутників і наземних станцій, що забезпечує глобальне визначення місця розташування, швидкості та часу за допомогою радіосигналів;

- INS (Inertial Navigation System) – це автономна система, яка визначає положення, швидкість і орієнтацію об'єкта за допомогою гіроскопів і акселерометрів без зовнішніх сигналів. Включає у себе магнітometri – сенсори, що вимірюють магнітне поле для визначення орієнтації апарата відносно магнітного північного полюса. Та барометричні висотоміри – пристрої, що визначають висоту над рівнем моря шляхом вимірювання атмосферного тиску;

- VBN (Vision-Based Navigation) – це система навігації, що використовує камери для визначення положення та орієнтації об'єкта за

аналізом зображень навколишнього середовища, входить до підсистем комп'ютерного зору для навігації;

- VDM (Vehicle Dynamic Model) – це математична модель, яка описує поведінку транспортного засобу в різних умовах руху. Модель використовує фізичні рівняння для моделювання сили, прискорення, швидкості, орієнтації та інших характеристик транспортного засобу в реальному часі.

Блок моніторингу несправностей виконує функції інтеграційного фільтра, що об'єднує дані з різних джерел для отримання точних оцінок поточного стану системи. Цей блок відіграє критичну роль у виявленні несправностей та забезпеченні стабільності роботи системи в реальному часі. Серед доступних підходів до інтеграції даних одним із найбільш поширених і ефективних є фільтр Калмана.

Фільтр Калмана являє собою рекурсивний цифровий алгоритм, що дозволяє прогнозувати новий вектор стану системи на основі двох ключових компонентів: моделі динаміки системи та даних, отриманих із сенсорів або інших джерел. Його робота базується на оцінюванні стану в два етапи: прогнозування та корекція. Під час етапу прогнозування алгоритм використовує математичну модель для передбачення майбутнього стану системи. На етапі корекції це передбачення уточнюється з урахуванням нових вимірювань, що надходять у реальному часі.

Головна перевага фільтра Калмана полягає в його здатності працювати з шумними даними та враховувати невизначеності вхідних сигналів. Завдяки цьому він забезпечує стабільні та точні оцінки стану навіть у складних умовах експлуатації, таких як динамічні зміни в середовищі або несподівані збої сенсорів.

Існують модифікації фільтра Калмана, зокрема розширений фільтр Калмана (EKF) та несамоспрямований фільтр Калмана (UKF), які використовуються для нелінійних систем. Ці модифікації дозволяють застосовувати фільтр у ширшому спектрі задач, включно з робототехнікою, автономними транспортними засобами, авіацією та космічними

технологіями.

Загалом, використання інтеграційного фільтра, такого як фільтр Калмана, у блоці моніторингу несправностей дозволяє своєчасно виявляти аномалії в роботі системи, знижувати ризики відмов і забезпечувати її ефективне функціонування в режимі реального часу [5].

Блок об'єднання представляє собою обробку уніфікованих даних з різних підсистем та датчиків та є автономним центром прийняття рішень. Компонент працює як автономний центр прийняття рішень, оскільки на основі узгоджених даних формує команди для інших підсистем, зокрема для навігації, уникнення перешкод і виконання заданого маршруту. Такий підхід дозволяє забезпечити ефективну взаємодію між компонентами системи та підвищити точність прийняття рішень в умовах мінливої динаміки або перешкод.

У системах безпілотних літальних апаратів за стандартом UTM – P, V і A позначають наступне:

- 1) P – Position (Позиція) – просторове розташування безпілота в тривимірному просторі;
- 2) V – Velocity (Швидкість) – швидкість та напрям руху БПЛА, виражається у вигляді вектора з компонентами по осях x , y і z ;
- 3) A – Acceleration (Прискорення) – позначає зміну швидкості з часом, може бути лінійною (зміна швидкості в прямолінійному русі) або кутовою (зміна кутової швидкості). кутовим прискоренням (зміна швидкості обертання).

2.2 Перспективи автономної навігації та штучний інтелект

На сьогоднішній день існує безліч рішень із використанням штучного інтелекту для автономної навігації БПЛА в різноманітних умовах.

Комп'ютерне бачення – це метод вилучення значення з візуального введення; це робиться в режимі реального часу на БПЛА, щоб сприймати їх

робоче середовище. Революція штучного інтелекту доповнила цей розвиток, дозволивши використовувати найсучасніші підходи CV (Computer Vision) у сферах, які ніколи не вважалися можливими, наприклад, досягнення автономності під час дозаправки в повітрі. Загалом, навігаційні моделі CV навчаються з використанням підходу до навчання з перенесенням, коли навчання проводиться у віртуальному 3D-середовищі, яке імітує умови роботи БПЛА, а потім навчання розгортається на фізичній машині. Цей метод знижує вартість аварії, пов'язаної з тестуванням у реальному світі.

Іншою перспективною технікою штучного інтелекту для керування автономними роботами є навчання з підкріпленням. Reinforcement Learning чи RL базується на навчанні агента через взаємодію із навколишнім середовищем. Процес такого навчання складається із декількох етапів, що виконуються ітеративно поки модель не починає показувати бажані результати:

- 1 агент із датчиків отримує до входу дані про поточний стан середовища;
- 2 стратегія, що є, зазвичай складною, математичною функцією, приймає рішення на основі отриманих даних та обирає одну чи декілька можливих для роботи дій;
- 3 агент застосовує обрану дію у середовищі;
- 4 на основі факторів успішності виконання завдання, наприклад відстань до кінцевої точки, вираховується нагорода за виконану дію;
- 5 агент оновлює стратегію згідно із отриманою нагородою та новим станом.

Даний метод дозволяє досягти успішного виконання завдання у різних умовах завдяки ітерації вище зазначених кроків.

Також необхідно виділити недоліки навчання з підкріпленням:

- розмір, варіабельність і репрезентативність навчальних наборів даних. Так як складність будь-якого модуля ШІ для навігації швидко зростає разом зі складністю завдання, яке потрібно виконати. Наприклад, для аналізу

короткострокових змін похибки інерційних датчиків за допомогою нейронної мережі потребується кілька шарів обробки, тоді як для навчання повного конвеєра візуальної одометрії потрібна дуже глибока мережа з різними типами нейронів [6];

- доступність наборів даних. Більшість методів на основі ШІ дуже чутливі до якості та різноманітності навчальної вибірки. Відсутність різноманітності в навчальних вибірках призводить до нездатності навченої моделі узагальнювати невидимі дані. Основна частина збору даних у спільноті дослідників ШІ спрямована на автономне водіння. Як наслідок, у відкритому доступі є лише декілька наборів аерофотоданих з достатньо великими розмірами вибірок, щоб покрити репрезентативну зону польоту в контексті БПЛА.

Отже, моделі, навчені на даних, зібраних в цих умовах, не будуть узагальнюватись для передбачуваної операційної області. Потенційним рішенням для прискорення глибокого навчання в БПЛА є використання симуляцій як джерела навчальних даних. Тестування навченої моделі потім виконується в реальних експериментах [7].

2.3 Існуючі реалізації

Український навігаційний комплекс «Бавовна» являє собою революційну розробку в галузі автономної навігації, що поєднує в собі передові досягнення у сфері штучного інтелекту та сучасних сенсорних технологій – представлено на рисунку 2.2.

Основу комплексу становить модульна архітектура, яка дозволяє легко адаптувати його до різних платформ, включно з безпілотними літальними апаратами, наземними роботами та іншими автономними системами. До ключових мультисенсорних компонентів належать: інерційні сенсори високої точності для вимірювання прискорень та кутових швидкостей, лідари та камери для просторового сканування та побудови карт місцевості в

реальному часі, GNSS.



Рисунок 2.2 – Навігаційний комплекс "Бавовна"

Обчислювальні властивості комплексу представляють систему із високопродуктивних процесорів із низьким енергоспоживанням, інтеграційних графічних процесорів для прискорення обробки даних із сенсорів та реалізації алгоритмів глибокого навчання, підтримку FPGA.

Навігаційна система «Бавовна» позиціонується як додаткова навігаційна система з технологією штучного інтелекту, що дозволяє виконувати місію через зони, де заборонено GPS та загрозу радіоелектронної боротьби, водночас пропонуючи вдосконалений інтерфейс на основі відкритої платформи Ardupilot.

Як зазначає виробник, основними сферами застосування комплексу «Бавовна» є: військові операції, цивільна авіація, робототехніка [8].

Високий технічний потенціал навігаційного комплексу «Бавовна» дозволяє адаптуватися до різноманітних умов експлуатації, забезпечувати точність і надійність у динамічному середовищі, а також демонструє прогрес у розвитку українських технологій у сфері автономної навігації.

3 АЛГОРИТМИ АВТОНОМНОЇ НАВІГАЦІЇ

3.1 Проблематика та актуальність у сфері безпілотних літальних апаратів

Автономна навігація вважається одним із найскладніших питань у розробці безпілотних літальних апаратів. Створення системи навігації, що базується на декількох алгоритмах орієнтування, у тому числі застосування штучного інтелекту, породжує проблему вибору оптимальної траєкторії безпілота та, як наслідок, – затримки обробки даних.

Ще однією значущою проблемою є невизначеність середовища. У реальних умовах дрони стикаються з різноманітними перешкодами та змінними умовами, такими як погодні явища, неочікувані об'єкти або швидка зміна рельєфу. Ландшафтні особливості, такі як ліс, пустеля, урбанізовані території, гірська місцевість, суттєво ускладнюють планування маршрутів.

Для вирішення цих викликів використовуються наступні підходи:

- динамічні алгоритми орієнтування на місцевості, які здатні адаптувати маршрут у реальному часі;
- навчання з підкріпленням для створення моделей, що можуть самостійно вдосконалювати свої навички орієнтування на основі досвіду;
- методи побудови дерева перешкод, що забезпечують ефективний пошук траєкторій у складних середовищах.

Важливим аспектом у розробці автономних систем є баланс між універсальністю та ефективністю. Універсальні рішення часто потребують значних обчислювальних ресурсів, що може бути неприйнятним для легких дронів із обмеженими технічними можливостями. Тому більш практичним підходом стає використання планувальників маршрутів, оптимізованих для конкретних типів середовища або завдань. Наприклад, алгоритми, що враховують попередньо завантажені карти місцевості, дозволяють значно

знизити обчислювальні витрати та забезпечити швидку реакцію на зміни.

Отже, використання адаптивних алгоритмів і навчання з підкріпленням дозволяє значно підвищити ефективність автономної навігації, забезпечуючи гнучкість у подоланні перешкод та змін середовища.

Орієнтація та навігація є критично важливими аспектами для безпілотних літальних апаратів, особливо при роботі в умовах невизначеного навігаційного середовища. Вибір оптимального маршруту та забезпечення коректної орієнтації під час польоту стають ключовими завданнями для забезпечення ефективності та надійності роботи таких систем. Далі будуть розглянуті як традиційні методи, які фокусуються на прямолінійних польотах та статичних корекціях, так і комплексні алгоритми.

3.2 Алгоритм SLAM

SLAM (Simultaneous Localization and Mapping) – це процес, за допомогою якого мобільний робот може побудувати карту навколишнього середовища й водночас використовувати цю карту для визначення свого місцезнаходження. У SLAM як траєкторія платформи, так і розташування всіх орієнтирів оцінюються безперервно без необхідності будь-яких апріорних знань про місцезнаходження [9]. Алгоритм працює з сенсорами: камерами, LiDAR, IMU та поділяється на 2D (плоскі карти) та 3D (просторові карти).

Проблематика алгоритму SLAM полягає у декількох причинах:

- 1 шуми сенсорів. Дані, отримані з сенсорів (камер, LiDAR, IMU тощо), містять шуми. Це ускладнює точне визначення положення робота та особливостей середовища;
- 2 проблеми зі збігом даних (Data Association) У SLAM важливо правильно зіставляти особливості, виявлені сенсорами, між кількома кадрами. Помилки у зіставленні можуть призводити до неправильного визначення положення або побудови карти;

- 3 дрейф позиції (Drift). У випадках, коли SLAM працює без зовнішніх орієнтирів, таких як GNSS, відбувається накопичення помилок через інерційні вимірювання або внутрішню оцінку траєкторії;
- 4 обмежені обчислювальні ресурси. SLAM вимагає високих обчислювальних ресурсів для виконання алгоритмів локалізації та побудови карти в режимі реального часу. Це стає викликом для роботів із обмеженими апаратними можливостями;
- 5 відсутність надійного закриття циклу (Loop Closure). Для забезпечення глобальної узгодженості карти необхідно ефективно розпізнавати раніше відвідані місця. Помилки в цьому процесі призводять до зсуву карти та поганої узгодженості;
- 6 обмеження у великих або динамічних середовищах. SLAM стикається з викликами у великих масштабах або в умовах, де об'єкти постійно змінюють свої позиції (наприклад, у натовпі людей чи на транспортному потоці);
- 7 складність реалізації в реальному часі. Обчислювальні процеси SLAM, такі як розв'язання великих систем рівнянь, графова оптимізація та зменшення похибок, є дорогими за часом. Це створює труднощі для роботи в реальному часі.

3.3 Візуальна одометрія

Візуальна одометрія – це техніка комп'ютерного бачення для оцінки пози та руху камери на основі послідовності зображень і є фундаментальною можливістю, необхідною для комп'ютерного зору та автономних навігаційних роботів. Використовуючи камери споживчого рівня, а не дорогі датчики чи системи, підхід є недорогим і простим для оцінки місцезнаходження роботів і транспортних засобів. Цей метод забезпечує поступову оцінку положення робота у реальному часі шляхом зіставлення послідовних кадрів зображення в послідовності та накопичення відносних

поз між кадрами. На відміну від технології SLAM, візуальна одометрія не підтримує карту навколишнього середовища, що дозволяє роботам локалізувати та здійснювати навігацію без запису спостережуваних ключових точок. Дана технологія характеризується хорошим балансом між складністю реалізації, надійністю та вартістю. Метод може ефективно працювати в середовищах, де сигнал GPS слабкий або відсутній [6]. Нагальну демонстрацію роботи алгоритму візуальної одометрії було використано із дослідження [10] рисунок 3.1:

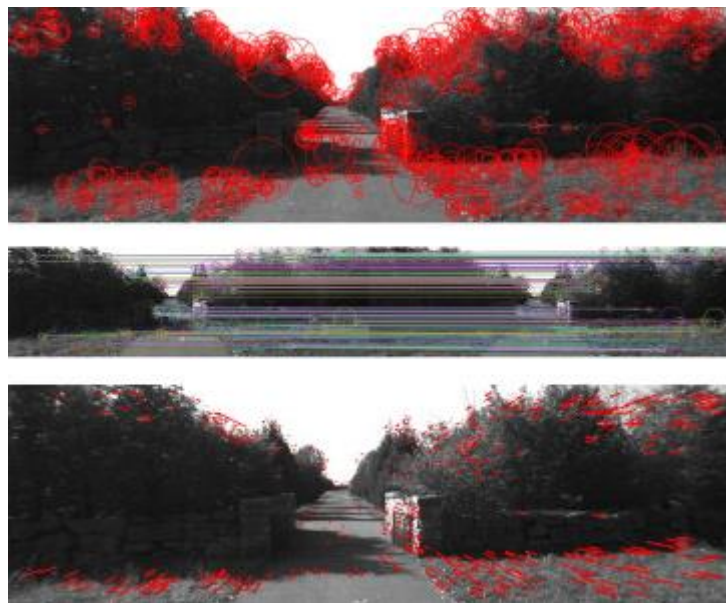


Рисунок 3.1– Демонстрація роботи алгоритму візуальної одометрії

3.4 Алгоритми планування траєкторії

Проблеми планування шляху та траєкторії є центральними у галузі робототехніки та автоматизації. Сучасні тенденції розвитку цих напрямків спрямовані на досягнення високих швидкостей роботи, що дозволяє зменшити час виконання виробничих операцій. Проте збільшення швидкості може негативно впливати на точність і повторюваність рухів роботів через високі вимоги до виконавчих механізмів та систем керування.

Одним із ключових аспектів ефективного функціонування робототехнічних систем є створення траєкторій, які забезпечують високу швидкість виконання завдань при мінімізації впливу надмірних прискорень та вібрацій на механічну конструкцію. Такі траєкторії повинні бути плавними, щоб уникати небажаних динамічних ефектів, що виникають через різкі зміни швидкості чи прискорення.

Алгоритми планування шляху генерують геометричну траєкторію від початкової до кінцевої точки, яка може проходити через заздалегідь визначені проміжні точки в робочому просторі. Натомість алгоритми планування траєкторії накладають часові параметри на визначений геометричний шлях, що є важливим для кінематичних і динамічних характеристик руху.

Алгоритми планування шляху поділяють на кілька основних категорій залежно від методологій, які використовуються для побудови геометричного шляху:

- 1 методи дорожньої карти: передбачають побудову графів для визначення оптимального шляху;
- 2 алгоритми декомпозиції клітинок: розбивають робочий простір на менші області для пошуку траєкторії;
- 3 методи штучного потенціалу: застосовують силові поля для направлення робота до цілі, уникаючи перешкод;
- 4 алгоритми планування траєкторії зазвичай класифікують за цільовою функцією, яку необхідно оптимізувати.

Основними підходами є:

- мінімізація часу: забезпечує найшвидше виконання завдання;
- мінімізація енергії: зменшує енергоспоживання системи;
- мінімізація ривка: сприяє зниженню динамічних впливів та вібрацій.

Крім того, у науковій літературі описані гібридні алгоритми, які одночасно оптимізують кілька цільових функцій, наприклад, поєднання мінімізації часу та енергії.

Таким чином, дослідження в області планування шляху та траєкторії мають велике значення для розвитку робототехніки. Вивчення та удосконалення цих алгоритмів дозволяє створювати більш ефективні, швидкі та надійні робототехнічні системи.

3.4.1 Класичний метод. A*

Алгоритм A* (A-star) – це пошуковий алгоритм, широко застосовуваний для задач пошуку шляхів, особливо в області робототехніки. A* належить до класу інформованих пошукових алгоритмів, тобто він використовує евристичну функцію для оцінки відстані до мети.

Його основна ідея полягає в тому, щоб спланувати мінімальний шлях пересування без зіткнень від початкової позиції до цільової позиції на даній карті. Він обчислює пріоритет кожного вузла за допомогою функції вартості, представлені формулою (3.1):

$$f(n)=g(n)+h(n) \quad (3.1)$$

де $f(n)$ – це загальна оцінка вартості від початкового стану через стан n до цільового стану, і чим вище значення, тим нижчий пріоритет;

$g(n)$ – фактична вартість від початкового стану до стану n ;

$h(n)$ – це оцінена вартість оптимального шляху від стану n до цільового стану, також відома як евристична вартість алгоритму A*.

Традиційна реалізація алгоритму A* поступово старіє, проте його модифіковані версії залишаються актуальними для сучасних задач, особливо за умов оптимізації для конкретних сценаріїв, як для безпілотних літальних апаратів.

Для подолання обмежень A* розроблено модифікації, такі як:

- Hierarchical Pathfinding A*: працює швидше на великих мапах за рахунок використання ієрархій;

- Bidirectional A*: досліджує шляхи одночасно від початкової точки і мети;
- Iterative Deepening A*: зменшує використання пам'яті, використовуючи ітераційне заглиблення [11].

3.4.2 Статистичний метод Rapidly-Exploring Random Trees

Класичний алгоритм RRT (Rapidly-Exploring Random Tree) працює шляхом поступового розширення дерева в просторі конфігурацій, спрямованого на пошук траєкторії між початковою і цільовою точками. Алгоритм починає з кореня дерева, що відповідає початковій конфігурації. Далі в просторі конфігурацій генерується випадкова точка. Алгоритм шукає вузол у дереві, найближчий до цієї випадкової точки, і розширює дерево в цьому напрямку на невелику фіксовану відстань, створюючи новий вузол.

Новий вузол додається до дерева лише тоді, якщо шлях до нього не перетинає перешкоди. Якщо новий вузол наближається до цільової точки, алгоритм перевіряє можливість прямого з'єднання з нею. Коли таке з'єднання можливе, шлях вважається знайденим. Якщо ж кількість ітерацій перевищує встановлений ліміт, алгоритм завершується без результату [12].

RRT чудово працює у складних просторах, де створення повної карти середовища є непрактичним. Але результуючі траєкторії часто є неоптимальними, а дослідження простору – нерівномірним, що згодом стимулювало розвиток його вдосконалених версій.

4 РОЗРОБКА АЛГОРИТМУ

4.1 Виклики забезпечення надійності навігаційних систем

Гарантування безпеки (стосовно навігації) в авіації залежить від надзвичайно суворої перевірки потенційних режимів відмов і експлуатаційних меж систем, що надають навігаційні дані. Тому чіткою вимогою є передбачувана і зрозуміла людині алгоритмічна логіка. Однією з ключових переваг традиційного підходу до навігації на основі аналітичних моделей є те, що критичні параметри роботи системи можна передбачити для будь-якої заданої конфігурації системи і набору умов навколишнього середовища. Однак найбільша перевага ІШ, а саме його здатність до навчання і адаптації, є одночасно і найбільшим його недоліком, оскільки логічна адаптація в польоті не може бути формально перевірена заздалегідь. Це призводить до відсутності відстеження між отриманим алгоритмічним дизайном і основним процесом, що моделюється. Тому задля безпеки автономного управління роботом, постає задача комплексної розробки таких методів. Де першим етапом є залучення інженерних навичок людини, написання алгоритму, а другий – навчання за допомогою ІШ, методом підбору ваг та з чітким контролем з боку людини.

4.2 Алгоритм 3D-VFH

Векторна гістограма поля (VFH) є алгоритмом планування руху в реальному часі, запропонованим Йоганном Боренштейном та Йорамом Кореном у 1991 році. Метод базується на статистичному представленні навколишнього середовища через сітку гістограми, що дозволяє ефективно враховувати невизначеності, зумовлені сенсорними даними та помилками моделювання. На відміну від багатьох інших алгоритмів уникнення

перешкод, VFH враховує як динаміку, так і форму робота, генеруючи керуючі команди, адаптовані до конкретної платформи [13].

3D-VFH алгоритм, що далі буде розглядатись – це розширена версія двовимірного алгоритму VFH який використовується для роботи у тривимірному середовищі. Алгоритм використовує гістору для того, щоб визначити, де знаходяться перешкоди, враховуючи положення робота в 3D-середовищі. Алгоритм 3DVFH використовує п'ять етапів для розрахунку нового руху робота:

1 сканування середовища – сенсори дрона (лазерні далекоміри або камери) збирають інформацію про об'єкти навколо, створюючи тривимірну карту простору в оперативній пам'яті бортового комп'ютера;

2 побудова гістограми – усі бар'єри перетворюються на вектори, які вказують напрямок і відстань до них. Сама гістограма представлена у пам'яті системи як набір секторів, кожен з яких описує можливий вектор безпілота, заповнений значеннями небезпеки (відсутність чи близькість бар'єра);

3 аналіз секторів – алгоритм шукає серед секторів ті, де найменше завад, щоб вибрати безпечну траєкторію апарату. Чим менше значення в секторі гістограми, тим безпечніше переміщення у цьому напрямку;

4 прийняття рішення – робот виконує переміщення у напрямку з мінімальним ризиком, балансуючи між досягненням цілі та униканням обмежувальних чинників.

Однак, не зважаючи на високу точність роботи алгоритму 3D-VFH в статичному просторі, він втрачає свою ефективність для керування БПЛА в умовах, що змінюються. Основним недоліком цього методу є обмеження у прогнозуванні поведінки перешкод зі змінними характеристиками. Метод орієнтований на локальні рішення, що знижує ефективність у складних сценаріях з високою швидкістю польоту, де потрібно заздалегідь оцінювати траєкторію руху робота та змінних факторів, що обмежують рух.

Одним із найефективніших покращень підходу 3D-VFH для БПЛА в динамічно змінюваних обставинах є інтеграція методу ймовірнісного планування шляхів. Використання моделі на основі ймовірнісних графів дозволяє передбачати маршрути об'єктів, що змінюють своє положення, і відповідно коригувати маршрут літального апарата. Це дозволяє знизити ризик зіткнень і покращити обчислювальну ефективність.

Замість миттєвого аналізу середовища алгоритм ймовірнісного планування траєкторій прогнозує переміщення на основі моделей поведінки. Ця система дозволяє забезпечити баланс між безпечністю маневрування, його плавністю та ефективністю за витратами часу й енергії. Дана технологія являє собою багаторівневий процес з інтеграцією A* алгоритму:

1 вибір цілі – безпілотник обирає безпечну точку для планування, враховуючи ймовірність існування обмежень;

2 дискретний пошук – використовується розширений A*-пошук для мінімізації ризику зіткнень зі статичними та динамічними об'єктами, моделюючи їх як ймовірнісні графи й векторні поля швидкостей;

3 згладжування траєкторії – результати пошуку перетворюються на безперервний маршрут з обмеженням на зіткнення;

4 перевірка коректності – здійснюється контроль динамічних обмежень і плавність руху БПЛА.

Ефективність алгоритму ймовірнісного планування траєкторій було продемонстровано у симуляціях і тестах на безпілотниках. Порівняно з традиційними методами, цей підхід забезпечує до 3.8 разів вищу успішність уникнення перешкод і використовує лише на 18% більше часу, необхідного для базової стратегії 3D-VFH. Дане дослідження було апробовано та представлено на науковій конференції під темою «Потенціал алгоритму 3D Vector Field Histogram в автономних літальних апаратах» [14].

Хоча класична реалізація 3D-VFH демонструє високу ефективність у статичних обставинах її застосування в динамічних середовищах, обмежене через неспроможність прогнозувати рух бар'єрів. Водночас вдосконалення

цього підходу, зокрема інтеграція методів ймовірнісного планування траєкторій, відкриває значний потенціал для розв'язання цієї проблеми. Поєднання локального аналізу та прогнозування змін з високою варіативністю дозволяє адаптувати маршрут роботи до складних сценаріїв, забезпечуючи вищу точність і надійність навігації. Таким чином, покращені версії 3D-VFH мають перспективу стати ефективними рішеннями для автономної навігації літальних апаратів в умовах реального світу.

4.3 Середовище моделювання та технічні характеристики

Для моделювання поведінки безпілотного летального апарату було вирішено використати комп'ютерну симуляцію у тривимірному просторі, в якому робот має можливість рухатися відповідно до заданого вектору із вказаною швидкістю. Даного дослідження спрямоване на виявлення особливостей та вдосконалення навігаційного алгоритму 3D-VFH задля ефективної роботи у динамічно змінюваному середовищі, тож в програмній симуляції не будуть враховуватись такі фактори, як недосконалості контролерів та датчиків конкретних моделей БПЛА.

Всі експерименти та випробування проводяться на комп'ютері із наступними характеристиками:

- процесор: Intel(R) i7-8700K @3.70GHz;
- графічний адаптер: NVIDIA GeForce GTX 1650, 4ГБ відеопам'яті;
- кількість оперативної пам'яті: 16 ГБ;
- операційна система: Windows 10.

На сьогоднішній день існує безліч додатків, що дозволяють виконувати симуляції у тривимірному просторі. Для дослідження, що проводиться, було розглянуто такі програми, як Unreal Engine, Unity, Blender, Gazebo та Autodesk Maya. Для того щоб зробити вибір відповідного середовища моделювання, було сформовано список вимог, яким має задовольняти обране рішення:

- 1 наявність зручних та багатофункціональних інструментів програмування поведінки об'єктів у 3D-просторі;
- 2 можливість безкоштовного користування додатком для дослідницьких цілей;
- 3 наявність фізичного рушія, що дає змогу симулювати фізичні процеси, які будуть максимально наближені до реального світу;
- 4 відповідність технічних вимог програми наявній апаратурі.

Після аналізу наявних рішень для моделювання експериментів у тривимірному просторі, було обрано платформу Unity, тому що вона відповідає усім зазначеним вище критеріям.

Unity – інструмент для розроблення відеоігор і застосунків, і рушій, на якому вони працюють. Створені за допомогою Unity програми можуть виконуватися на настільних комп'ютерних системах, мобільних пристроях та гральних консолях у дво- та тривимірній графіці. Програмне забезпечення найчастіше використовується для розробки комп'ютерних і мобільних ігор, але також активно застосовується в інших галузях, таких як симуляції, навчальні програми, візуалізація фізичних процесів та робототехніка.

Вбудований у Unity фізичний рушій PhysX забезпечує реалістичну симуляцію таких фізичних процесів, як гравітація, динамічні зіткнення, тертя та аеродинамічний опір. Він може дозволити відтворення поведінки безпілотного літального апарату в максимально наближених до реальності умовах. Крім цього, користувач має змогу налаштування фізичних параметрів, змінюючи характеристики середовища, коефіцієнти тертя та опору повітря, масу чи центр ваги об'єктів. Зазначені можливості сприяють кращому розумінню впливу різних факторів на параметри навігації, траєкторію польоту та покращення стабільності руху БПЛА в умовах, що змінюються.

Unity пропонує широкі можливості для створення спеціальних сценаріїв поведінки об'єктів у тривимірному просторі завдяки потужному API та підтримці мови програмування C#. Також платформа надає

можливість задавати власні алгоритми руху літальних апаратів, інтегруючи їх із симуляцією перешкод, змінних погодних умов та постійних змін у навколишньому середовищі. Потужність у створенні подібних сценаріїв дає змогу відтворити різноманіття реальних ситуацій для тестування навігаційних алгоритмів та забезпечення детального моделювання ітерацій для покращення автономної поведінки дронів.

Ці ключові переваги роблять Unity універсальним інструментом для досліджень у сфері навігації автономних роботів та безпілотних летальних апаратів, особливо в ситуаціях коли ресурси для експериментів у реальному світі обмежені.

4.4 Модифікація алгоритму 3D-VFH для покращення уникнення перешкод

Для забезпечення руху робота максимально близько до бажаної траєкторії d з одночасним уникненням зіткнень запропоновано реальний час планування траєкторій. Метод передбачає створення довгих траєкторій, які робот виконує протягом короткого часу, а потім повторно планує у наступній ітерації. Передбачається, що системи сприйняття, прогнозування та оцінювання стану працюють незалежно від планувальника, забезпечуючи його необхідною інформацією.

Статичні перешкоди: множина O , яка містить опуклі форми перешкод та ймовірність їхнього існування $p_{static}(Q)$, де $Q \in O$.

Динамічні перешкоди: множина D , яка включає:

- поточні позиції перешкод $p_D^{current}$;
- функції форми зіткнення R^D ;
- моделі поведінки $m_{D,i}$ із відповідними ймовірностями реалізації $p_{dynamic}(D, i)$;
- сукупність станів $S=\{s_0, ..., s_c\}$, де кожен стан описує поточне положення та інші параметри робота.

Алгоритм складається з чотирьох основних етапів (рисунок 4.1):

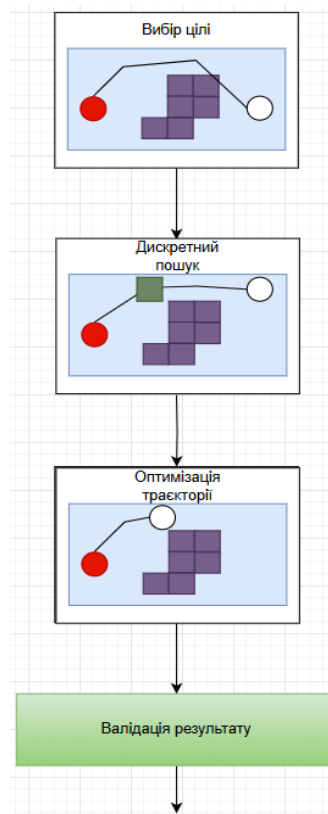


Рисунок 4.1 – Схематичне представлення етапів роботи алгоритму

Можна виділити чотири етапи формування траєкторії руху безпілотної в поточному стані. Дана стратегія працює в динамічних умовах, а також є ймовірнісною безпечною:

- 1 вибір цілі: визначення точки на бажаній траєкторії d і часу її досягнення;
- 2 дискретний пошук: обчислення дискретного шляху у просторі-часі до обраної цілі з мінімізацією ймовірностей зіткнень;
- 3 оптимізація траєкторії: згладжування дискретного шляху для забезпечення безпеки руху;
- 4 валідація результату: оцінка відповідності траєкторії динамічним обмеженням роботи.

У разі невдачі під час оптимізації або перевірки дійсності роботи

продовжує використовувати попередній план. Якщо планування успішне, траєкторія оновлюється.

4.5 Вибір цілі та математична постановка задачі

Далі буде розглянуто стадію вибору цілі в алгоритмі, що визначає, куди має рухатися робот. Вибір включає визначення точки g на бажаній траєкторії d , до якої планується рух, і моменту часу T_0 , коли ця точка має бути досягнута.

Ключові моменти, які варто виділити:

1) параметри вибору цілі:

- горизонт планування (τ): інтервал часу, на який здійснюється прогнозування;

- ймовірність існування статичної перешкоди ($p_{\min}^O$): мінімальна ймовірність, за якої статична перешкода враховується при плануванні.

2) механізм вибору цілі:

- алгоритм шукає точку часу T_0 , найближчу до $\tilde{T} + \tau$, де $\tilde{T}$ — поточний момент часу;

- у момент T_0 робот, який перебуває на бажаній траєкторії d , повинен уникати зіткнень із усіма статичними перешкодами $Q \in O$, для яких ймовірність існування $p_{\text{static}}(Q) \geq p_{\min}^O$.

3) особливості процесу:

- вибір здійснюється лише для однієї точки на бажаній траєкторії d , яка є безпечною з точки зору уникнення перешкод;

- Реальну траєкторію робота визначає подальша робота алгоритму.

4) формалізація задачі:

- описує математичну постановку задачі, яка конкретизує умови та обмеження вибору цілі.

У стадії вибору цілі визначається положення g на бажаній траєкторії d , до якого здійснюється планування, а також момент часу T_0 , у який це

положення повинно бути (або повинно було бути) досягнуто.

Стадія вибору цілі складається з двох параметрів – бажаний горизонт планування (τ), мінімальна ймовірність існування статичної перешкоди ($p_{\min}^O$).

Нехай $\tilde{T}$ – поточний момент часу. Алгоритм вибору цілі визначає момент часу T_0 , який є найближчим до $\tilde{T} + \tau$ (тобто до моменту часу, що знаходиться на бажаному горизонті планування від поточного часу), за умови, що робот, перебуваючи на бажаній траєкторії у момент T_0 , буде поза зоною зіткнень із усіма статичними перешкодами $Q \in O$, для яких виконується умова: $p_{\text{static}}(Q) \geq p_{\min}^O$.

Варто зауважити, що вибір цілі здійснюється лише для однієї точки на бажаній траєкторії d , яка є безпечною з точки зору уникнення перешкод. Фактична траєкторія, якою рухатиметься робот, буде спланована іншими етапами алгоритму [15].

Формально, задача вибору цілі на цій стадії формуються наступним чином у рівнянні 4.1.

Знайти:

$$T_0 = \arg \min_{T \in \mathcal{T}} |T - (\tilde{T} + \tau)| \quad (4.1)$$

Де $\mathcal{T}$ – множина моментів часу, за яких виконується умова 4.2:

$$p_{\text{static}}(Q) \geq p_{\min}^O \quad \forall Q \in O. \quad (4.2)$$

Вибір цілі у кодї представлено у лістингу 4.1, що викликається при кожному оновленні позиції літального апарата.

Лістинг 4.1 – фрагмент перевірки знаходження безпілотної на траєкторії до цільової точки

```
private bool IsAtDestination()
{
    return Vector3.Distance(transform.position,
targetDestination) <= destinationThreshold;
}
```

На стадії дискретного пошуку планується шлях до цільової позиції g із використанням алгебраїчного пошуку вартості A^* . Алгебраїчний пошук A^* є узагальненням стандартного пошуку A^* для більш багатой множини функцій вартості, так званих алгебраїчних структур вартості.

Алгебра вартості – це математична структура $(A, \wedge, \times, \leq, 0, 1)$, де:

- A – множина значень витрат;
- $\wedge$ – операція вибору найменшої витрати;
- $\times$ – операція накопичення витрат;
- $\leq$ – відношення тотального порядку;
- 0 і 1 – відповідно найгірше і найкраще значення витрат.

Для підтримки кількох критеріїв під час пошуку використовують пріоритетний декартовий добуток алгебраїчних структур вартості, який дозволяє порівнювати пари витрат за заданими правилами пріоритетності.

Алгоритм A^* на основі цієї моделі виконує пошук з урахуванням пріоритетів і сукупних витрат, забезпечуючи знаходження оптимального шляху за заданими критеріями [15].

Дискретний пошук по секторам продемонстровано у даному лістингу 4.2:

Лістинг 4.2 – Дискретно-секторний пошук найоптимальнішого шляху

```
private Vector3 GetDirectionFromSectors(int hSector, int
vSector)
{
    float horizontalAngle = hSector * (360f /
horizontalSectors) * Mathf.Deg2Rad;
    float verticalAngle = vSector * (180f / (verticalSectors
```

```

- 1)) * Mathf.Deg2Rad - Mathf.PI / 2;
    float x = Mathf.Cos(verticalAngle) *
Mathf.Cos(horizontalAngle);
    float y = Mathf.Sin(verticalAngle);
    float z = Mathf.Cos(verticalAngle) *
Mathf.Sin(horizontalAngle);

    return new Vector3(x, y, z).normalized;
}

```

У процесі експериментального дослідження встановлено, що ефективність оптимізації алгоритму ґрунтується на дискретно-секторному пошуку. Варто зазначити, що адаптація алгебраїчних структур вартості може здійснюватися безпосередньо в реальних польових умовах, що забезпечує можливість отримання валідаційних даних для подальшого вдосконалення алгоритму. Балансування цих структур із використанням методів штучного інтелекту дозволяє досягти високої прозорості результатів як для людини, так і для системи ШІ, що є суттєвою перевагою в контексті інтерактивного навчання та оптимізації.

На етапі оптимізації траєкторії обмеження на максимальні значення похідних враховуються лише в дискретно вибраних точках, однак отримана траєкторія може залишатися динамічно недопустимою між цими точками.

На етапі перевірки динамічної допустимості перевіряється, чи задовольняються максимальні величини похідних γ_k для $k \in \{1, \dots, K\}$, тобто в умові 4.3:

$$\left\| \frac{d^k f(t)}{dt^k} \right\|_2 \leq \gamma_k, \forall k \in \{1, \dots, K\} \quad (4.3)$$

Якщо траєкторія не задовольняє цю умову, вона відкидається, і ітерація планування вважається невдалою.

Якщо ж траєкторія $f(t)$ проходить перевірку, вона вважається допустимою, оскільки відповідає максимальним величинам похідних, які робот здатний виконати, і є безперервною до ступеня k , що було забезпечено на етапі оптимізації траєкторії. Ці умови є необхідними для динамічної

допустимості траєкторії для роботів із диференційною пласкістю [13]. Перевірка динамічної допустимості траєкторії продемонстрована у реалізації у методі `CalculateBestDirection()` визначає найкращий напрямок руху, враховуючи вартісні функції (обхід перешкод, вирівнювання до цілі, врахування імпульсу, передбачення динамічних перешкод). `PredictDynamicObstacleCost()`: прогнозує можливість зіткнення з динамічними перешкодами.

4.6 Аналіз якості алгоритму

Для наочного випробування алгоритму 3D-VFH розроблену програмну реалізацію було інтегровано в середовище Unity. Використання Unity дозволило створити віртуальну симуляцію, що максимально наближена до реальних умов експлуатації безпілотних літальних апаратів, представлено на рисунку 4.2.

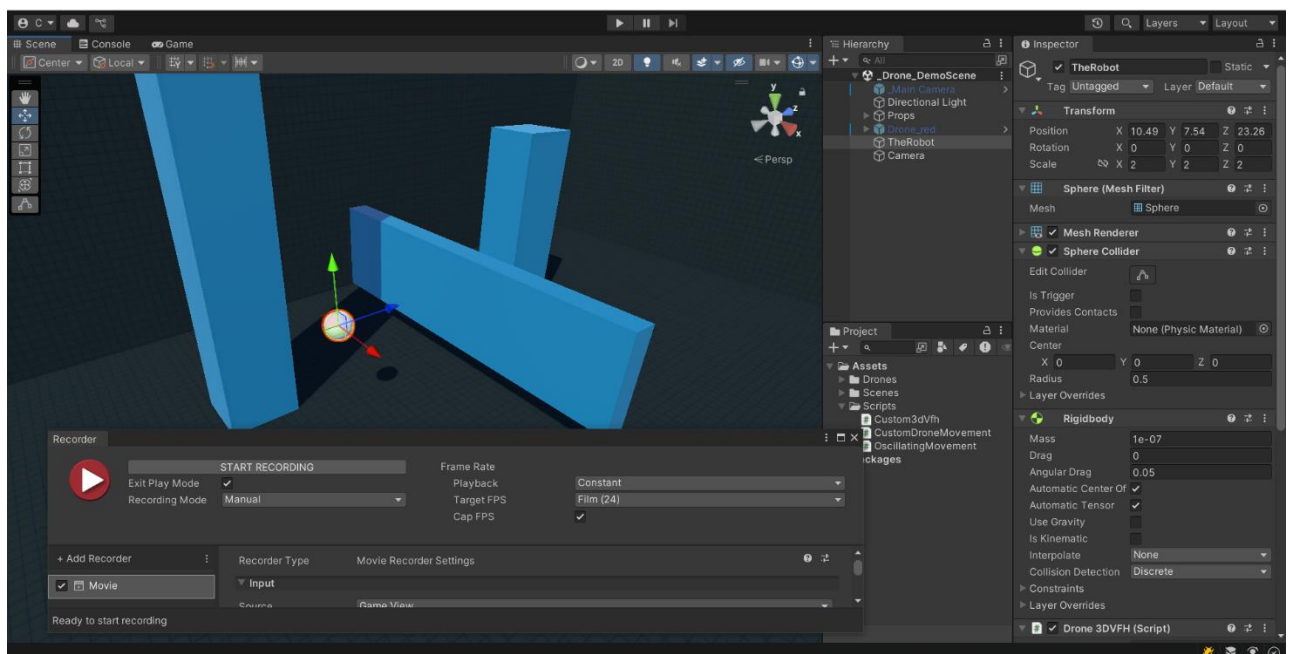


Рисунок 4.2 – Середовище розробки Unity

Задля тестування алгоритму 3D-VFH із ймовірнісним плануванням

траєкторій на ефективність дії та виявлення долі помилок у подоланні перешкод, була створена програмна симуляція, що імітує динамічні перешкоди, що мають довільне розташування, розміри, напрямки та швидкість руху.

Об'єкти (компоненти) тестування включали:

- тестове середовище – це тривимірний віртуальний простір, створений для проведення експериментального тестування. На рисунку 4.3 представлено це середовище, яке включає об'єкти для імітації перешкод, визначення траєкторій та інших елементів, необхідних для оцінки ефективності алгоритму 3D-VFH;

- дрон – це спрощений віртуальний об'єкт у формі сфери, який використовується для спрощення моделювання та аналізу. Така геометрія обрана через її симетричність і легкість обчислень у контексті тестування алгоритмів. Об'єкт моделює основні фізичні властивості дрона, зокрема інерцію, рух у тривимірному просторі та реакцію на перешкоди. Для зручності аналізу дрон оснащений візуальними індикаторами:

- синій промінь – позначає напрямки руху;
- червоні промені – відображають дискретно-секторний пошук 3D-VFH алгоритму, демонструючи обробку навколишніх даних. А також напрямки, які вважаються безпечними або небезпечними (рисунки 4.4 а);

- точка призначення – червона сфера, що позначає кінцевий пункт маршруту (рисунки 4.4 б);

- стіна – об'єкт, який виконує роль фізичної перешкоди, що блокує рух дрона у визначених зонах тривимірного простору. У процесі дослідження моделюються як статичні перешкоди, що залишаються нерухомими протягом усього польоту, так і динамічні перешкоди, які змінюють своє положення з часом, симулюючи реальні сценарії зіткнення з рухомими об'єктами або змінним середовищем. Це дозволяє перевірити адаптивність алгоритму до різних умов навігації та оцінити його здатність до швидкої реакції на зміну ситуації (рисунки 4.5).

Такий підхід до тестування дозволяє наочно продемонструвати ефективність алгоритму 3D-VFH у різних сценаріях. Завдяки можливостям Unity можна моделювати адаптацію алгоритму до змін середовища та уникнення перешкод, що підвищує практичну цінність дослідження.

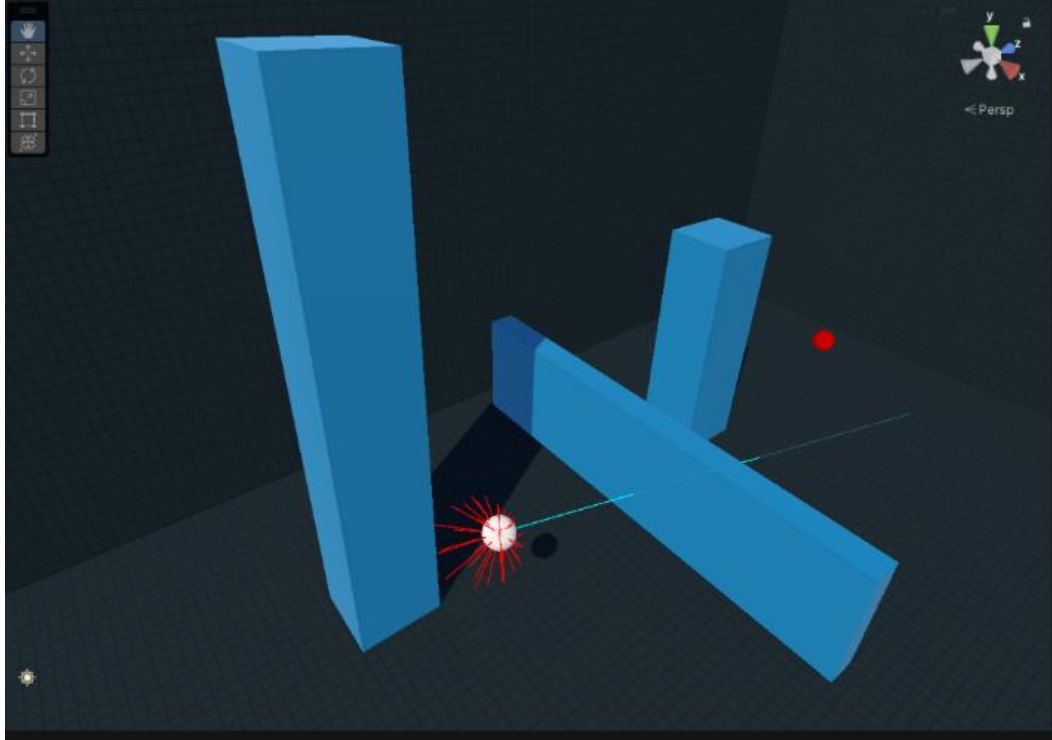


Рисунок 4.3 – Тестове середовище

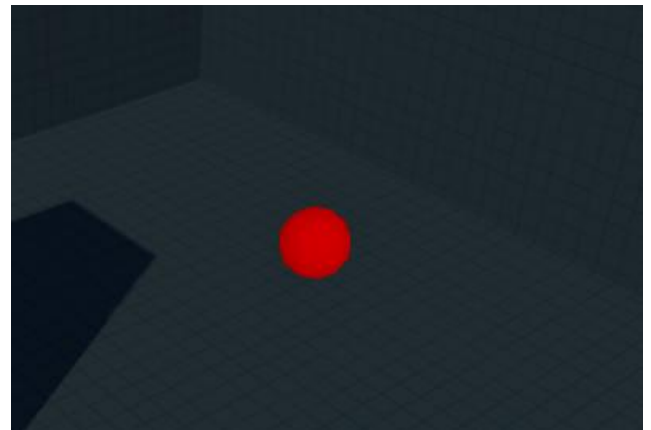
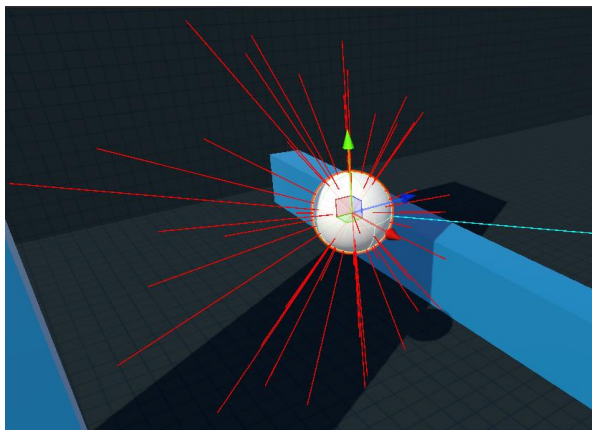


Рисунок 4.4 – Об'єкти тестування у віртуальному просторі: а) дрон – об'єкт у вигляді кулі; б) точка прибуття – червона куля, що вказує на фінальний пункт

призначення



Рисунок 4.5 – Фізична перешкода «стіна»

Симуляція має такі характеристики:

- розмір простору: 100x100x50 метрів;
- кількість динамічних перешкод: від 10 до 20 одиниць що дозволяє змодельовати різноманітні сценарії з різним ступенем складності;
- розміри перешкод: мінімальна довжина ребра паралелепіпеду від 0.5 до 15 метрів;
- швидкість руху перешкод: від 0 до 15 метрів в секунду;
- максимальна швидкість руху автономного робота: 10 метрів в секунду;
- відстань від початкового положення автономного робота до точки призначення: 90 метрів.

Результати проведених випробувань представлені в таблиці 4.1.

Таблиця 4.1 – Оцінка факторів виробничого середовища і процесу

№	Час подолання (сек)	Час просто ю (сек)	Оптимальність траєкторії (%)	Блокування (так/ні)	Успішність (так/ні)	Кількість зіткнень
1	12.4	1.8	105	Ні	Так	0
2	14.2	2.3	110	Ні	Так	0
3	15.8	3.1	112	Ні	Так	0
4	16.4	0.5	107	Ні	Так	0
5	13.7	1.1	106	Ні	Так	0
6	18.9	4.0	118	Ні	Так	0
7	-	-	-	Так	Ні	2
8	14.8	2.6	111	Ні	Так	0
9	12.6	1.5	104	Ні	Так	0
10	13.9	1.7	108	Ні	Так	0
11	16.1	3.0	115	Ні	Так	0
12	-	-	-	Так	Ні	0
13	14.0	2.0	109	Ні	Так	0
14	12.3	1.2	103	Ні	Так	0
15	-	-	-	Ні	Ні	1
16	14.5	2.4	108	Ні	Так	0
17	13.2	0.9	106	Ні	Так	0
18	15.0	2.5	112	Ні	Так	0
19	18.1	4.1	117	Ні	Так	0
20	19.7	4.5	119	Ні	Так	0
21	12.9	1.3	105	Ні	Так	0
22	14.6	2.2	111	Ні	Так	0
23	17.8	3.8	116	Ні	Так	0
24	-	-	-	Так	Ні	0

Продолження таблиці 4.1

№	Час подолання (сек)	Час простою (сек)	Оптимальність траєкторії (%)	Блокування (так/ні)	Успішність (так/ні)	Кількість зіткнень
25	13.4	1.6	106	Ні	Так	0
26	19.2	4.2	118	Ні	Ні	1
27	20.9	5.5	121	Ні	Так	0
39	16.0	2.9	114	Ні	Так	0
40	19.6	4.4	120	Ні	Так	0
41	17.4	3.5	116	Ні	Так	0
42	14.4	2.2	108	Ні	Так	0
43	20.7	6.0	123	Ні	Ні	2
44	15.6	2.8	113	Ні	Так	0
45	18.0	3.7	117	Ні	Так	0
46	21.9	6.5	125	Ні	Так	0
47	14.3	2.1	109	Ні	Так	0
48	19.0	4.2	118	Ні	Так	0
49	12.8	1.3	104	Ні	Так	0
50	15.2	2.6	110	Ні	Так	0
51	14.1	2.0	109	Ні	Так	0
52	15.7	2.9	113	Ні	Так	0
53	18.5	3.9	117	Ні	Так	1
54	13.1	1.0	105	Ні	Так	0

Усього було проведено 54 ітерацій симуляції роботи алгоритму в динамічних умовах. Під час кожного випробування було зафіксовано результати роботи автономного робота за наступними характеристиками:

- час подолання маршруту – час, за який робот подолав маршрут від свого початкового положення до точки призначення;
- час простою (зупинок) – час, коли робот був змушений зупинитися або різко знизити швидкість для планування нової траєкторії;
- оптимальність траєкторії – відхилення фактичної пройденої відстані від прямої між початковою та кінцевою точками;

- кількість зіткнень із перешкодами;
- чи було зафіксоване блокування робота;
- успішність виконання завдання – кількість успішних спроб подолання маршруту без зіткнення з перешкодами чи блокування.

Таким чином, експериментально було визначено що:

- середній час проходження маршруту: 16.14 секунди;
- середній час простою: 2.97 секунди;
- середня оптимальність траєкторії: 112.53%;
- доля успішних спроб подолати маршрут: 88%;
- доля блокування: 6%;
- доля зіткнень із перешкодами: 10%.

Результати експерименту показують, що алгоритм 3D-VFH із ймовірнісним плануванням траєкторій досягає хорошої продуктивності в середовищі з динамічними перешкодами. Середній час виконання вказує на помірну затримку під час планування нових траєкторій для уникнення перешкод. У той же час середня оптимальність маршруту на рівні 112,53% вказує на певне відхилення від найкоротшого шляху, що є особливістю алгоритмів, заснованих на безпеці автономного руху.

Частка успішних спроб подолати збудований маршрут становить 88%, що хоч і є високою, але залишає місце для оптимізації. Виявлене блокування в 6% і зіткнення з перешкодами в 10% вказує на необхідність удосконалення алгоритму, щоб зробити його більш стійким до складних динамічних ситуацій.

Загалом, алгоритм має хороші перспективи для використання в автономних навігаційних системах, хоча необхідна подальша оптимізація, направлена на зменшення кількості зіткнень і підвищення точності траєкторії.

4.7 Порівняння з іншими алгоритмами

Результати досліджень алгоритму 3D-VFH демонструють його ефективність у задачах уникнення перешкод у динамічних середовищах завдяки швидкому аналізу даних із сенсорів та адаптації траєкторії у реальному часі. Проте для забезпечення точного визначення положення дрона, у вузьких проходах або довгих перельотах, можливостей цього алгоритму недостатньо. У таких випадках корисним є порівняння 3D-VFH із візуальною одометрією, яка має суттєво інший підхід до обробки даних.

Візуальна одометрія та метод 3D-VFH суттєво відрізняються за своїм підходом до вирішення задач автономної навігації дронів, оскільки орієнтовані на різні аспекти керування та аналізу простору. Візуальна одометрія ґрунтується на обробці послідовності зображень, отриманих із камер (моно- або стереокамер), з метою визначення переміщення дрона у просторі. Однак цей метод залежить від зовнішніх умов, таких як освітленість і текстурованість середовища, що може знижувати його ефективність у несприятливих умовах.

На відміну від візуальної одометрії, метод 3D-VFH базується на даних сенсорів, які вимірюють відстань до об'єктів у просторі, зокрема лідара, ультразвукових сенсорів або стереокамер. Як показало дослідження, метод демонструє високу ефективність у динамічному середовищі, де наявні рухомі об'єкти, оскільки дозволяє швидко адаптуватися до змін у просторі. Водночас його обмеження проявляються у складних середовищах із вузькими проходами або за недостатньої якості даних від сенсорів.

Вибір між візуальною одометрією та 3D-VFH залежить від конкретних завдань. Для локалізації та картографування, особливо в умовах, де відсутнє GNSS-покриття, візуальна одометрія є оптимальним вибором. Водночас для уникнення перешкод у реальному часі, особливо у динамічних середовищах, метод 3D-VFH демонструє кращі результати. Цей факт пояснює доцільність дослідження саме 3D-VFH для вирішення завдань навігації безпілотних літальних апаратів в умовах активної взаємодії з перешкодами та необхідності швидкої реакції на зміни в оточенні.

Для комплексних задач та забезпечення стабільності безпілотного літального апарату, найбільш ефективним підходом є інтеграція обох методів. Візуальна одометрія забезпечує точну локалізацію та відстеження положення дрона, тоді як 3D-VFH відповідає за безпечне планування траєкторії та уникнення перешкод. Така комбінація дозволяє забезпечити високу адаптивність і надійність системи у складних умовах експлуатації.

ВИСНОВКИ

У процесі роботи було здійснено огляд сучасних типів БПЛА, їх класифікацію, а також аналіз їх ролі у військових і цивільних операціях. Це підкреслює актуальність досліджень у сфері інтелектуального керування, включаючи адаптивні методи на основі підкріплювального навчання. Застосування штучного інтелекту продемонструвало переваги для забезпечення адаптації дронів до змін середовища, підвищення їхньої здатності до автономного прийняття рішень і оптимального планування траєкторій у реальному часі.

Окрім цього, було досліджено методи захисту БПЛА від зовнішніх впливів, таких як радіоелектронна боротьба (РЕБ) та кібернетичні атаки. Продемонстровано важливість використання інерційних навігаційних систем як засобу протидії блокуванню сигналів GNSS, що забезпечує надійну роботу дронів у складних умовах.

У межах проведеного дослідження розглянуто важливу наукову задачу, пов'язану з удосконаленням методів автономного керування безпілотними літальними апаратами в динамічних умовах. Особливу увагу приділено алгоритмам уникнення перешкод, серед яких тривимірна гістограма векторних полів – алгоритм 3D-VFH. Проведені симуляції підтвердили, що застосування алгоритму 3D-VFH дозволяє досягати високої успішності уникнення перешкод завдяки аналізу локальних просторових даних. Важливим аспектом є інтеграція системи алгебраїчних структур вартості, що дозволяє порівнювати витрати на основі заданих правил пріоритетності. Такий підхід надає можливість адаптувати алгоритм до різних умов середовища, забезпечуючи високу точність уникнення перешкод.

Додатково система алгебраїчних структур вартості відкриває перспективу навчання алгоритму за допомогою штучного інтелекту із впровадженням контрольованості з боку людини. Це забезпечує не тільки

автономність, але й можливість інтерактивного налаштування параметрів для виконання конкретних завдань, що значно підвищує універсальність та ефективність застосування таких систем.

Завдяки порівнянню різних підходів до уникнення перешкод визначено їхні сильні та слабкі сторони. Метод 3D-VFH забезпечує баланс між ефективністю та обчислювальними витратами, що є критично важливим для застосування в оперативних бойових умовах. Результати дослідження власної реалізації алгоритму показують, що статистика успішних спроб подолати побудований маршрут становить 88%. Це значення є високим, але вказує на потенціал для подальшої оптимізації.

Проведені дослідження й експерименти сприяють створенню вискооефективних автономних систем керування для безпілотників. Отримані результати мають потенціал для подальшого використання в оборонній сфері та цивільному секторі, особливо для виконання складних завдань у динамічних та невизначених середовищах.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. О.І. Тимочко Д.Ю. Голубничий В.Ф. Третьяк І.В. Рубан. Класифікація БПЛА за призначенням. Системи озброєння і військова техніка. 2007. № 1(9). URL: <https://openarchive.nure.ua/server/api/core/bitstreams/878899d8-b7a7-4481-af22-9835c0748ba0/content> (дата звернення: 20.01.2025).
2. STANAG 4670 MINIMUM TRAINING REQUIREMENTS FOR UNMANNED AIRCRAFT SYSTEMS (UAS) OPERATORS AND PILOTS. Status: active Effective from 2025-01-20. Official edition. Brussels, 2019. 20 p.
3. K Guthrie M. United States Department of the Army. www.army.mil. URL: https://www.army.mil/article/198368/auds_links_electronic_warfare_and_ammunition (дата звернення: 20.01.2025).
4. Federal Aviation Administration. UAS Traffic Management. 2023. 13 p. URL: https://www.faa.gov/uas/research_development/traffic_management/UTM-Key-Site-Ops-Eval-Workshop.pdf (дата звернення: 20.01.2025).
5. Jain S. K., Singh S. Harmonics estimation in emerging power system: Key issues and challenges. Electric Power Systems Research. 2011. Issue: 9, Volume: 81. URL: <https://www.sciencedirect.com/science/article/abs/pii/S0378779611001088> (дата звернення: 20.01.2025).
6. Scaramuzza D., Fraundorfer F. Visual Odometry Part I: The First 30 Years and Fundamentals. IEEE ROBOTICS & AUTOMATION MAGAZIN. 2011 (дата звернення: 20.01.2025).
7. Bijjahalli S., Gardi A., Sabatini R. Advances in Intelligent Autonomous Navigation Systems for Small UAS. Aerospace Sciences. 2020. Volume: 115. URL: <https://doi.org/10.1016/j.paerosci.2020.100617> (дата звернення: 20.01.2025).
8. Гібридна інерціальна навігація (H-INS) з використанням AI для UV під загрозами заборони GPS / EW. bavovna.ai. URL: <https://bavovna.ai/ai->

- navigation-kit/ (дата звернення: 20.01.2025).
9. Durrant-Whyte H., Bailey T. Simultaneous Localization and Mapping: Part I. IEEE ROBOTICS & AUTOMATION MAGAZIN. 2006. P. 99 (дата звернення: 20.01.2025).
 10. Evaluation of non-geometric methods for visual odometry / T. A. Ciarfuglia та ін. Robotics and Autonomous Systems. 2014. Т. 62, № 12. С. 1717–1730. URL: <https://doi.org/10.1016/j.robot.2014.08.001> (дата звернення: 20.01.2025).
 11. A Systematic Literature Review of A* Pathfinding / D. Foead et al. Procedia Computer Science. 2021. Vol. 179. P. 507–514. URL: <https://doi.org/10.1016/j.procs.2021.01.034> (дата звернення: 20.01.2025).
 12. A rapidly exploring random tree optimization algorithm for space robotic manipulators guided by obstacle avoidance independent potential field / X. Gao et al. International Journal of Advanced Robotic Systems. 2018. Vol. 15, no. 3. P. 172988141878224. URL: <https://doi.org/10.1177/1729881418782240> (дата звернення: 20.01.2025).
 13. Borenstein J., Koren Y. The vector field histogram-fast obstacle avoidance for mobile robots. IEEE Transactions on Robotics and Automation. 1991. Vol. 7, no. 3. P. 278–288. URL: <https://doi.org/10.1109/70.88137> (дата звернення: 20.01.2025).
 14. К.Ю. Сорока. «Потенціал алгоритму 3D Vector Field Histogram в автономних літальних апаратах». Current Trends In Scientific Research Development, Boston , 2025 (дата звернення: 20.01.2025).
 15. IROS 2020 CFP. IEEE Robotics & Automation Magazine. 2020. Vol. 27, no. 1. P. 13. URL: <https://doi.org/10.1109/mra.2020.2974123> (дата звернення: 20.01.2025).