

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет імені Петра Могили
Факультет комп'ютерних наук
Кафедра інтелектуальних інформаційних систем

ДОПУЩЕНО ДО ЗАХИСТУ

В. о. завідувача кафедри інтелектуальних
інформаційних систем

_____ Євген СІДЕНКО

« ____ » _____ 2025 р.

КВАЛІФІКАЦІЙНА РОБОТА
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ МАГІСТРА
ІНТЕЛЕКТУАЛЬНА СИСТЕМА КЕРУВАННЯ
ДОСТАВКОЮ ВАНТАЖІВ В ДИНАМІЧНОМУ
СЕРЕДОВИЩІ ЗА ДОПОМОГОЮ БПЛА НА БАЗІ UNITY

Спеціальність 122 Комп'ютерні науки

Освітня програма «Інтелектуальні інформаційні системи»

Здобувач

_____ Олексій Фоменко

« ____ » _____ 2025 р.

« ____ » _____ 2025 р.

Керівник д-р техн. наук, професор

_____ Олексій КОЗЛОВ

« ____ » _____ 2025 р.

Миколаїв – 2025

Чорноморський національний університет імені Петра Могили
(повне найменування закладу вищої освіти)

Факультет	Комп'ютерних наук
Кафедра	Інтелектуальних інформаційних систем
Рівень вищої освіти	Другий (магістерський)
Освітній ступень	Магістр
Спеціальність	122 Комп'ютерні науки
Освітня програма	Інтелектуальні інформаційні системи

ЗАТВЕРДЖУЮ

В. о. завідувача кафедри інтелектуальних
інформаційних систем

_____ Євген СІДЕНКО

« ____ » _____ 2025 р.

ЗАВДАННЯ
на кваліфікаційну роботу здобувача

Фоменка Олексія Вікторовича

((прізвище, ім'я, по батькові здобувача))

1. Тема кваліфікаційної роботи «Інтелектуальна система керування доставкою вантажів в динамічному середовищі за допомогою БПЛА на базі Unity».
Керівник роботи Козлов Олексій Валерійович, д-р техн. наук, професор.
Затверджена наказом ЧНУ ім. Петра Могили від «07» липня 2025 р. № 184.
2. Строк представлення кваліфікаційної роботи «16» грудня 2025 р.
3. Очікуваний результат роботи та початкові дані, якщо такі потрібні: параметри навколишнього середовища, моделі маршрутів та логістичних точок, алгоритми навігації, дані для симуляції динамічних умов, критерії оцінки ефективності.
4. Перелік питань, що підлягають розробці: аналіз сучасного стану розвитку систем автономного керування БПЛА; огляд та порівняння наявних алгоритмів навігації та уникнення перешкод; вибір та обґрунтування алгоритмів,

використаних для керування рухом БПЛА; формування наборів параметрів і характеристик, необхідних для побудови моделей навігації та керування; розробка інтелектуальної системи керування доставкою вантажів в динамічному середовищі за допомогою БПЛА на базі Unity; тестування та оцінка ефективності розробленої системи.

5. Перелік графічних матеріалів: презентація.

Керівник роботи

(Особистий підпис)

Олексій КОЗЛОВ
(Власне ім'я ПРІЗВИЩЕ)

Здобувач

(Особистий підпис)

Олексій ФОМЕНКО
(Власне ім'я ПРІЗВИЩЕ)

Дата видачі завдання «28» червня 2025 р.

КАЛЕНДАРНИЙ ПЛАН

кваліфікаційної роботи

Тема: Інтелектуальна система керування доставкою вантажів в динамічному середовищі за допомогою БПЛА на базі Unity

№	Найменування роботи	Початок	Закінчення	Примітки
1	Отримання завдання на виконання КР	27.06.2025	30.06.2025	Виконано
2	Аналіз предметної області та постановка задачі	1.07.2025	20.07.2025	Виконано
3	Огляд літературних джерел за темою кваліфікаційної роботи	21.07.2025	30.07.2025	Виконано
4	Огляд наявних систем керування БПЛА для вирішення поставленої задачі	01.09.2025	25.10.2025	Виконано
5	Реалізація обраних технологій з аналізом отриманих результатів	26.10.2025	21.11.2025	Виконано
6	Перший попередній захист КР на засіданні комісії кафедри	25.11.2025	26.11.2025	Виконано
7	Корегування роботи за результатами попереднього захисту	27.11.2025	05.12.2025	Виконано
8	Другий попередній захист КР на засіданні комісії кафедри	09.12.2025	09.12.2025	Виконано
9	Доробка та остаточне оформлення КР	10.12.2025	12.12.2025	Виконано
10	Подання КР, її електронної копії та інших документів (відгуку, рецензії) до захисту	15.12.2025	16.12.2025	Виконано

Керівник роботи

(Особистий підпис)

Олексій КОЗЛОВ
(Власне ім'я ПРІЗВИЩЕ)

Здобувач

(Особистий підпис)

Олексій ФОМЕНКО
(Власне ім'я ПРІЗВИЩЕ)

Дата складання календарного плану
«08» липня 2025 р.

АНОТАЦІЯ

кваліфікаційної роботи студента групи 601 ЧНУ ім. Петра Могили

Фоменка Олексія Вікторовича

Тема: «Інтелектуальна система керування доставкою вантажів в динамічному середовищі за допомогою БПЛА на базі Unity».

У сучасних логістичних системах усе більшої актуальності набуває застосування безпілотних літальних апаратів (БПЛА) для автономного доставлення вантажів, що дозволяє скорочувати час транспортування, зменшувати витрати та забезпечувати доступ до важкодоступних територій. Проте ефективне використання БПЛА у динамічному середовищі вимагає розробки інтелектуальних систем керування, здатних адаптувати маршрути, уникати перешкод, реагувати на зміни умов та забезпечувати стабільність польоту. У зв'язку з цим створення систем автономного управління дронами з використанням алгоритмів штучного інтелекту та засобів симуляції є важливим напрямом розвитку сучасних технологій автоматизації доставлення.

Об'єкт роботи – процеси керування рухом БПЛА під час виконання завдань доставлення вантажів у динамічному середовищі.

Предмет роботи – алгоритми навігації, оптимізації маршруту, уникнення перешкод та програмні моделі керування БПЛА, реалізовані на базі Unity.

Мета роботи полягає у розробці інтелектуальної системи керування доставленням вантажів із використанням БПЛА, здатної забезпечувати автономний політ, адаптацію до змін середовища, уникнення перешкод і ефективне виконання логістичних задач. Для досягнення мети застосовано алгоритми пошуку шляху, реактивної навігації, нечіткої логіки та моделі сенсорного аналізу, а симуляція середовища реалізована в Unity з використанням C#.

Пояснювальна записка складається зі вступу, чотирьох розділів, висновків та додатків.

У першому розділі проведено аналіз застосування БПЛА в логістичних системах, розглянуто особливості керування дронами у динамічному середовищі,

здійснено огляд сучасних алгоритмів навігації, планування маршрутів і методів уникнення колізій.

У другому розділі описано архітектуру інтелектуальної системи керування, її основні модулі, алгоритмічне забезпечення, моделі середовища та використані інструменти розробки, зокрема Unity, C# і засоби штучного інтелекту.

У третьому розділі наведено програмну реалізацію системи, включаючи структуру C#-скриптів, модулі керування БПЛА, систему уникнення перешкод, логіку взаємодії агентів і побудову сцени в середовищі Unity.

У четвертому розділі представлено результати тестування та візуалізації роботи системи, виконано аналіз поведінки одного та багатьох БПЛА, оцінено ефективність алгоритмів навігації, досліджено випадки колізій і механізми динамічного уникнення перешкод у різних сценаріях

У результаті роботи створено інтелектуальну систему керування доставленням вантажів за допомогою БПЛА, здатну виконувати автономну навігацію, реагувати на перешкоди, коригувати маршрут у реальному часі та забезпечувати стабільність польоту у змінних умовах. Система може бути використана для симуляцій, досліджень та подальшої інтеграції у реальні логістичні рішення.

Кваліфікаційна робота містить 80 сторінок, 16 рисунків, 3 таблиці, 58 джерело.

Ключові слова: безпілотні літальні апарати, автономна навігація, доставлення вантажів, Unity, алгоритми уникнення перешкод, A*, інтелектуальні системи керування.

ABSTRACT

**qualification work of a student of group 601 of BSNU named after Petro Mohyla
Fomenko Oleksii**

**Topic: "Intelligent cargo delivery management system in a dynamic environment
using UAVs based on Unity"**

In modern logistics systems, the use of unmanned aerial vehicles (UAVs) for autonomous cargo delivery is becoming increasingly relevant, which allows reducing transportation time, reducing costs and providing access to hard-to-reach areas. However, the effective use of UAVs in a dynamic environment requires the development of intelligent control systems that can adapt routes, avoid obstacles, respond to changing conditions and ensure flight stability. In this regard, the creation of autonomous drone control systems using artificial intelligence algorithms and simulation tools is an important direction in the development of modern delivery automation technologies.

The object of the work is the processes of controlling the movement of UAVs during the performance of cargo delivery tasks in a dynamic environment.

The subject of the work is navigation algorithms, route optimization, obstacle avoidance and software models of UAV control, implemented on the basis of Unity.

The purpose of the work is to develop an intelligent cargo delivery control system using UAVs, capable of providing autonomous flight, adaptation to environmental changes, obstacle avoidance and effective performance of logistics tasks. To achieve the goal, pathfinding algorithms, reactive navigation, fuzzy logic and sensor analysis models were used, and the environment simulation was implemented in Unity using C#.

The explanatory note consists of an introduction, four sections, conclusions and appendices.

The first section analyzes the use of UAVs in logistics systems, considers the features of drone control in a dynamic environment, reviews modern navigation algorithms, route planning and collision avoidance methods.

The second section describes the architecture of the intelligent control system, its main modules, algorithmic support, environment models and development tools used, in particular Unity, C# and artificial intelligence tools.

The third section presents the software implementation of the system, including the structure of C# scripts, UAV control modules, obstacle avoidance system, agent interaction logic and scene construction in the Unity environment.

The fourth section presents the results of testing and visualization of the system, analyzes the behavior of one and many UAVs, evaluates the effectiveness of navigation algorithms, investigates collision cases and dynamic obstacle avoidance mechanisms in various scenarios.

As a result of the work, an intelligent cargo delivery management system using UAVs was created, capable of performing autonomous navigation, reacting to obstacles, adjusting the route in real time and ensuring flight stability in changing conditions. The system can be used for simulations, research and further integration into real logistics solutions. The qualification work contains 80 pages, 16 figures, 3 tables, 58 sources.

Keywords: unmanned aerial vehicles, autonomous navigation, cargo delivery, Unity, obstacle avoidance algorithms, A*, intelligent control systems.

Кафедра інтелектуальних інформаційних систем
Інтелектуальна система керування доставкою вантажів в динамічному середовищі за допомогою БПЛА на
базі Unity

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ.....	3
ВСТУП.....	4
1 АНАЛІЗ ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАДАЧІ	6
1.1 Актуальність використання БПЛА для доставлення вантажів	6
1.2 Проблеми керування БПЛА в динамічному середовищі	8
1.3 Огляд сучасних систем і алгоритмів керування дронами.....	10
1.4 Аналіз аналогічних наявних систем керування доставкою вантажів за допомогою БПЛА	14
1.5 Методи оптимізації маршрутів і уникнення колізій.....	18
1.6 Постановка задачі дослідження.....	20
Висновки до розділу 1	22
2 СТРУКТУРА ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ КЕРУВАННЯ БПЛА	24
2.1 Архітектура системи.....	24
2.2 Алгоритми керування та навігації дронів.....	27
2.3 Моделі динамічного середовища та обробка перешкод.....	30
2.4 Використані інструменти та середовище розробки.....	32
3 РОЗРОБКА ТА РЕАЛІЗАЦІЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ.....	36
3.1 Алгоритм пошуку A^*	36
3.2 C#-файли та модулі Unity-проєкту.....	39
3.3 Система взаємодії модулів у Unity.....	42
3.4 Система зберігання даних та стану	45
3.5 Інтелектуальні алгоритми керування БПЛА	48
Висновки до розділу 3	52
4 ТЕСТУВАННЯ ТА ВІЗУАЛІЗАЦІЯ СИСТЕМИ.....	55
4.1 Тестування одного агента	55
4.2 Тестування кількох агентів	58
4.3 Тестування динамічного уникнення перешкод.....	62
Висновки до розділу 4	65
ВИСНОВКИ.....	68
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	70

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

БПЛА — безпілотний літальний апарат

DDPG — deep deterministic policy gradients

DQN — deep q-network

PID — proportional–integral–derivative

PSO — particle swarm optimization

SLAM — simultaneous localization and mapping

ВСТУП

Сучасний етап розвитку безпілотних літальних апаратів (БПЛА) характеризується їх активним впровадженням у різні галузі господарської діяльності, зокрема у логістику, транспортні системи, моніторинг територій та сферу доставки вантажів. Завдяки здатності здійснювати автономні польоти, оминати наземні перешкоди та працювати у складних умовах, БПЛА розглядаються як перспективна альтернатива традиційним засобам транспортування. Використання дронів для автоматизованої доставки дозволяє значно скоротити час транспортування, мінімізувати витрати на обслуговування транспортної інфраструктури, зменшити навантаження на дорожні мережі та забезпечити оперативний доступ до віддалених або важкодоступних регіонів.

Водночас ефективне застосування БПЛА в задачах доставки потребує розробки інтелектуальних систем керування, здатних функціонувати в умовах невизначеності та постійних змін навколишнього середовища. Такі системи повинні автономно планувати оптимальні маршрути, враховувати наявність статичних і динамічних перешкод, адаптуватися до змін погодних умов, а також координувати власні дії з іншими рухомими об'єктами, зокрема іншими безпілотними апаратами. Недостатній рівень інтелектуалізації керування може призводити до зниження ефективності доставки, зростання ризику зіткнень та порушення безпеки польотів.

Середовища моделювання реального часу, зокрема Unity, відкривають широкі можливості для проєктування, тестування та аналізу інтелектуальних систем керування БПЛА без необхідності використання реальних апаратів. Завдяки поєднанню фізичного рушія, гнучких інструментів візуалізації та підтримки програмування мовою C#, Unity дозволяє створювати складні інтерактивні сцени, у яких можливо відтворювати реалістичні сценарії польоту, моделювати поведінку перешкод і вплив зовнішніх факторів, а також перевіряти роботу алгоритмів у

режимі реального часу. Це робить Unity ефективною платформою для дослідження та апробації алгоритмів навігації, планування маршрутів і прийняття рішень.

Об'єкт роботи – процеси керування рухом БПЛА під час виконання завдань доставки вантажів у динамічному середовищі.

Предмет роботи – алгоритми навігації, оптимізації маршруту, уникнення перешкод та програмні моделі керування БПЛА, реалізовані на базі Unity.

Мета роботи полягає у розробці інтелектуальної системи керування доставкою вантажів із використанням БПЛА, здатної забезпечувати автономний політ, адаптацію до змін середовища, уникнення перешкод і ефективне виконання логістичних задач. Для досягнення мети застосовано алгоритми пошуку шляху, реактивної навігації, нечіткої логіки та моделі сенсорного аналізу, а симуляція середовища реалізована у Unity з використанням C#.

Для досягнення поставленої мети в роботі необхідно виконати такі основні завдання:

- провести аналіз сучасного стану проблеми керування БПЛА в умовах динамічного середовища;
- обґрунтувати вибір та здійснити порівняльний аналіз алгоритмів навігації й оптимізації траєкторії;
- розробити архітектуру інтелектуальної системи;
- реалізувати моделі навколишнього середовища;
- провести експериментальну оцінку розробленої системи.

У роботі проаналізовано сучасні підходи до побудови систем автономного керування, розглянуто реалізацію таких систем у Unity, описано архітектуру, основні компоненти, алгоритми навігації та методи оптимізації, а також порівняно класичні та інтелектуальні підходи до керування рухом БПЛА.

1 АНАЛІЗ ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Актуальність використання БПЛА для доставлення вантажів

Безпілотні літальні апарати (БПЛА, англ. UAV — Unmanned Aerial Vehicle) упродовж останнього десятиліття набули значного поширення у сфері логістики та транспортування вантажів, поступово перетворюючись на невід’ємний елемент сучасних інтелектуальних транспортних систем. Їх активне впровадження зумовлене стрімким розвитком технологій автономного керування, штучного інтелекту, супутникових і локальних навігаційних систем, комп’ютерного зору, а також підвищенням енергоефективності акумуляторних батарей. У сукупності ці фактори створили передумови для переходу від експериментального використання дронів до їх практичного застосування в реальних логістичних сценаріях.

Використання БПЛА дозволяє суттєво оптимізувати процеси доставки вантажів, особливо у випадках, коли традиційні транспортні засоби є економічно неефективними, повільними або фізично обмеженими у доступі до певних територій. Зокрема, дрони демонструють високу ефективність при доставці малогабаритних вантажів, медикаментів, термінових посилок, а також під час виконання завдань у сільській місцевості, гірських районах або густо забудованих міських зонах.

Серед основних переваг застосування БПЛА у логістичних системах доцільно виділити такі:

- зниження витрат на транспортування на коротких і середніх дистанціях завдяки економії пального або електроенергії, скороченню потреби у людських ресурсах та спрощенню транспортної інфраструктури;
- відсутність залежності від дорожньої мережі, що забезпечує можливість доставки вантажів у віддалені, важкодоступні або слаборозвинені регіони без додаткових витрат на будівництво шляхів сполучення;

- оперативність виконання доставки, особливо у критичних або надзвичайних ситуаціях, таких як природні катастрофи, техногенні аварії або зони підвищеної небезпеки;
- гнучке масштабування логістичних процесів, що дозволяє швидко змінювати кількість задіяних БПЛА залежно від інтенсивності замовлень, сезонних факторів чи потреб конкретного регіону;
- екологічну доцільність використання, оскільки більшість сучасних дронів оснащені електричними двигунами, що зменшує рівень викидів шкідливих речовин та шумове навантаження на довкілля.

Разом із тим, попри значний потенціал, широке впровадження БПЛА у сфері доставки вантажів супроводжується низкою складних технічних, організаційних та алгоритмічних проблем, які не можуть бути ефективно вирішені без застосування інтелектуальних методів керування. Серед ключових викликів доцільно виокремити такі:

- планування оптимального маршруту польоту, яке повинно враховувати обмеження енергоспоживання, масу вантажу, дальність польоту, погодні умови, рельєф місцевості та зони обмеженого або забороненого повітряного простору;
- уникнення зіткнень із перешкодами, до яких належать інші безпілотні апарати, будівлі, інженерні споруди, лінії електропередач, а також природні об'єкти, з використанням сенсорів, систем комп'ютерного зору та алгоритмів локальної навігації;
- функціонування у динамічному середовищі, де умови польоту можуть змінюватися в реальному часі під впливом поривів вітру, появи нових перешкод, зміни маршрутів або необхідності екстреного коригування траєкторії;
- координація багатоагентних систем, у яких одночасно працює велика кількість БПЛА, що виконують паралельні або взаємопов'язані завдання доставки, потребуючи узгодженого планування та мінімізації конфліктів у повітряному просторі.

Усі зазначені фактори формують об'єктивну потребу у створенні інтелектуальних систем керування, здатних комплексно аналізувати навколишнє середовище, прогнозувати потенційні ризики, приймати оптимальні рішення на основі поточних даних та адаптивно коригувати траєкторію руху БПЛА в режимі реального часу. Саме такі системи лежать в основі концепції «розумної логістики» та визначають подальші напрями розвитку автоматизованих систем доставки, орієнтованих на безпеку, ефективність і масштабованість у майбутніх транспортних екосистемах.

1.2 Проблеми керування БПЛА в динамічному середовищі

Керування безпілотними літальними апаратами (БПЛА) у динамічному середовищі характеризується низкою специфічних особливостей, які суттєво ускладнюють процеси навігації, планування маршруту та прийняття керувальних рішень у порівнянні з традиційними транспортними системами. Основна складність полягає в тому, що координати цілей, об'єктів та перешкод не є сталими, а змінюються в часі під впливом як зовнішніх, так і внутрішніх факторів. У зв'язку з цим система керування БПЛА повинна володіти високою адаптивністю та здатністю оперативно реагувати на зміни середовища.

Формування сигналів керування дроном здійснюється на основі поточних даних, що надходять від комплексу сенсорів, зокрема систем глобального позиціонування (GPS), інерціальних вимірювальних блоків (IMU), лідарів, барометрів та камер комп'ютерного зору. Ці сенсори забезпечують оцінку просторового положення апарата, його швидкості, висоти, орієнтації та напрямку руху. Важливою вимогою є обробка сенсорної інформації в режимі реального часу, оскільки навіть незначні затримки або похибки вимірювань можуть призвести до накопичення помилок, втрати стабільності польоту або зіткнення з перешкодами.

Додаткову складність для системи керування становить поява непередбачуваних ситуацій, до яких належать рухомі об'єкти (інші БПЛА, птахи,

транспортні засоби), раптові зміни погодних умов (пориви вітру, опади, зміна температури), поява нових статичних або динамічних перешкод, а також часткова або повна втрата каналу зв'язку з оператором чи наземною станцією керування. За таких умов БПЛА повинен бути здатним функціонувати автономно, підтримуючи безперервний зворотний зв'язок із середовищем та приймаючи рішення без участі людини.

Для забезпечення безпечного та ефективного руху в динамічному просторі використовуються інтелектуальні алгоритми прогнозування, які дозволяють оцінювати майбутні траєкторії інших об'єктів, визначати потенційні зони ризику та завчасно коригувати власний маршрут. Такі алгоритми можуть базуватися як на класичних методах математичного моделювання, так і на підходах машинного навчання та навчання з підкріпленням, що дає змогу системі накопичувати досвід і покращувати якість прийняття рішень з часом.

Типові завдання, які повинна вирішувати інтелектуальна система керування БПЛА у динамічному середовищі, включають:

- **побудову оптимального маршруту польоту**, що враховує енергоспоживання, тривалість місії, вантажопідйомність та вимоги до безпеки;
- **реактивне та проактивне ухилення від перешкод** у процесі руху з мінімальним відхиленням від запланованої траєкторії;
- **запобігання колізіям між кількома БПЛА** в спільному повітряному просторі шляхом координації дій та узгодженого планування;
- **адаптацію до змінних погодних умов**, зокрема компенсацію впливу вітру, опадів та температурних коливань на стабільність польоту;
- **автоматичне завершення місії та повернення на базу** у разі критичного зниження заряду акумулятора, втрати зв'язку або виникнення аварійних ситуацій.

Отже, керування безпілотними літальними апаратами в динамічному середовищі вимагає виконання складних обчислень у реальному часі, використання

алгоритмів штучного інтелекту, машинного навчання та методів прогнозування. Поєднання цих підходів дозволяє забезпечити стабільність, безпеку та високий рівень автономності польоту, що є ключовою передумовою ефективного застосування БПЛА у сучасних системах доставки та транспортної логістики.

1.3 Огляд сучасних систем і алгоритмів керування дронами

Розвиток автономних безпілотних літальних апаратів (БПЛА) є одним із ключових напрямів сучасних інтелектуальних технологій. Сучасні дрони широко застосовуються у військовій, транспортній, екологічній, аграрній та комерційній сферах. Основою їхньої ефективності є алгоритми навігації, стабілізації та ухвалення рішень у складному середовищі, що динамічно змінюється. Автономність БПЛА досягається завдяки інтеграції різних систем — сенсорних, обчислювальних і програмно-аналітичних модулів, які забезпечують можливість орієнтації в просторі, уникнення перешкод і планування оптимального маршруту.

Алгоритми керування дронами можна умовно поділити на чотири великі групи: класичні, евристичні, реактивні та інтелектуальні. Кожен підхід має свої переваги, обмеження та області застосування. У таблиці 1.1 наведено узагальнене порівняння основних методів навігації БПЛА.

Таблиця 1.1 – Порівняння основних алгоритмів навігації БПЛА

Тип алгоритму	Приклад / метод	Особливості	Переваги	Недоліки
Класичні	Dijkstra, A*	Використовують граф маршрутів	Простота реалізації, детермінованість	Погано адаптуються до змін середовища

Кінець таблиці 1.1

Тип алгоритму	Приклад / метод	Особливості	Переваги	Недоліки
Евристичні	Genetic Algorithm, Particle Swarm Optimization (PSO)	Використовують ітеративні пошуки мінімуму функції вартості	Підходять для складних задач	Висока обчислювальна складність
Реактивні	Potential Fields, Vector Field Histogram	Ухилення на основі локальної інформації	Реакція в реальному часі	Можливі "пастки локального мінімуму"
Інтелектуальні	Reinforcement Learning (Q-learning, DDPG)	Навчаються на прикладах або симуляціях	Висока адаптивність, здатність навчання	Необхідність великих даних та обчислень

Класичні алгоритми (наприклад, Dijkstra, A*) є основою для базового планування маршруту. Вони будують карту середовища у вигляді графа, де вузли відповідають точкам простору, а ребра — можливим шляхам руху. Їхня перевага — передбачуваність і гарантія знаходження оптимального маршруту, якщо він існує. Проте в умовах змінного середовища, де з'являються нові перешкоди або змінюється рельєф, такі методи працюють недостатньо ефективно, оскільки вимагають повного перерахунку маршруту.

Евристичні методи (зокрема Genetic Algorithm, PSO) застосовуються для пошуку оптимального рішення у випадках, коли точне обчислення є надто складним. Вони базуються на принципах біологічної еволюції або колективної

поведінки, що дозволяє ефективно розв'язувати задачі планування у складних просторових середовищах. Проте головним їхнім недоліком є потреба у великій кількості обчислень і, відповідно, часу, що обмежує їх використання у режимі реального часу.

Реактивні методи забезпечують швидке реагування на зміни середовища, використовуючи локальні сенсорні дані. Алгоритми на основі потенціальних полів або гістограм векторних полів дозволяють дрону оперативно ухилятися від перешкод без глобального планування маршруту. Це робить їх ефективними для польотів у непередбачуваних умовах, наприклад, у міських середовищах або під час польотів усередині будівель. Недоліком є можливість потрапляння у “локальні пастки” — ситуації, коли дрон не може знайти вихід із замкненого простору через відсутність глобального бачення карти.

Інтелектуальні методи становлять сучасний напрям розвитку автономного керування. Вони поєднують підходи машинного навчання, нечіткої логіки та глибоких нейронних мереж для формування моделей поведінки, які можуть самостійно вдосконалюватися на основі накопиченого досвіду або симуляцій. Такі алгоритми, як Reinforcement Learning (RL), дозволяють дрону навчатися шляхом проб і помилок, оптимізуючи дії для досягнення певної мети, наприклад, мінімізації енергоспоживання чи часу польоту.

Згідно з дослідженнями провідних наукових центрів — MIT, NASA Ames, найбільш перспективним напрямом є гібридні системи навігації, які комбінують декілька підходів одночасно. Такі системи дозволяють поєднати точність класичних методів з адаптивністю інтелектуальних.

До прикладів гібридних алгоритмів належать:

- A-RL (A-star + Reinforcement Learning) — спочатку дрон за допомогою A* знаходить базовий маршрут між початковою і кінцевою точками, після чого модуль підкріплювального навчання адаптує рух у реальному часі, реагуючи на появу нових перешкод або зміну погодних умов;

– PID + Fuzzy Control — поєднання класичного ПІД-регулятора для стабілізації руху з нечітким логічним коректором, який забезпечує плавну реакцію на зовнішні збурення (наприклад, пориви вітру або зміну навантаження). Така комбінація дає змогу досягти високої стабільності навіть за умов неповної або зашумленої інформації від сенсорів;

– DQN (Deer Q-Network) — один із найуспішніших підходів глибокого підкріплювального навчання, який дозволяє дрону навчатися уникати зіткнень і оптимізувати маршрут у тривимірному просторі без попереднього знання карти.

Крім цього, у сучасних системах керування активно розвивається напрям SLAM (Simultaneous Localization and Mapping), який дозволяє одночасно будувати карту місцевості та визначати поточне положення дрона. Інтеграція SLAM з глибинними нейронними мережами (Deer-SLAM) підвищує точність орієнтації в умовах обмеженої видимості або в середовищах без GPS.

Іншим важливим компонентом сучасних систем є використання злиття даних (sensor fusion). Це підхід, за якого інформація з різних сенсорів — камер, лідарів, ультразвукових далекомірів, інерційних вимірювальних блоків — об'єднується в єдину модель середовища. Застосування методів Баєсівського оцінювання, фільтра Калмана або фільтра частинок дозволяє суттєво зменшити похибки позиціонування та підвищити стабільність управління.

У контексті динамічного середовища, наприклад під час доставлення вантажів у міських умовах або в рятувальних операціях, найбільш ефективними виявляються саме гібридні та адаптивні системи, які здатні не лише реагувати на зміни, але й передбачати їх на основі попереднього досвіду. Поєднання моделей передбачення руху об'єктів з алгоритмами машинного навчання дозволяє мінімізувати ризик зіткнень і підвищити ефективність маршрутизації.

Таким чином, сучасні тенденції розвитку систем керування БПЛА спрямовані на інтеграцію інтелектуальних методів аналізу, евристичного пошуку, реактивних стратегій ухвалення рішень і адаптивних систем навчання. Розробка

подібних підходів є ключовим кроком до створення повністю автономних дронів, здатних самостійно діяти у складних і непередбачуваних умовах, що відкриває нові можливості для логістики, моніторингу та безпілотних перевезень у майбутньому.

1.4 Аналіз аналогічних наявних систем керування доставленням вантажів за допомогою БПЛА

Однією з найвідоміших промислових реалізацій автономного доставлення вантажів за допомогою безпілотних літальних апаратів є система Amazon Prime Air, яка передбачає транспортування малогабаритних посилок за допомогою мультикоптерів вертикального зльоту і посадки. Приклад такого апарата наведено на рис. 1.1. Дана система орієнтована на повністю автономне виконання місій доставлення на коротких дистанціях у межах міської та приміської інфраструктури.

Функціонування Amazon Prime Air базується на комплексному поєднанні GPS-навігації, систем комп'ютерного зору, лідарів та інших сенсорів уникнення перешкод, що забезпечують контроль просторового положення апарата та безпечне пересування у повітряному просторі. Особлива увага у проєкті приділяється питанням безпеки польотів, сертифікації апаратів і відповідності авіаційним стандартам. Водночас маршрути руху БПЛА формуються переважно у напівстатичному середовищі з жорстко визначеними повітряними коридорами та заздалегідь погодженими зонами польоту. Такий підхід значно спрощує контроль та регулювання польотів, однак обмежує гнучкість системи в умовах швидких змін навколишнього середовища або появи великої кількості одночасно активних агентів.



Рисунок 1.1 – Мультикоптер Amazon Prime Air

Схожі принципи автономного доставлення реалізовані у проєктах Wing (підрозділ компанії Google/Alphabet) та UPS Flight Forward, де БПЛА застосовуються для транспортування медикаментів, біологічних зразків та інших термінових вантажів. Приклади відповідних апаратів наведено на рис. 1.2. Дані системи характеризуються високим рівнем надійності, автоматизації та тісною інтеграцією з наявними авіаційними регуляторними системами, що дозволяє виконувати польоти в контрольованому повітряному просторі.

Разом із тим, такі рішення значною мірою залежать від централізованого керування, попереднього планування маршрутів і зовнішніх диспетчерських систем. Це ускладнює масштабування проєктів у динамічних середовищах, де необхідна оперативна адаптація маршрутів у реальному часі, а також координація великої кількості автономних БПЛА без постійного втручання оператора.

Кафедра інтелектуальних інформаційних систем
Інтелектуальна система керування доставкою вантажів в динамічному середовищі за допомогою БПЛА на базі Unity



a)



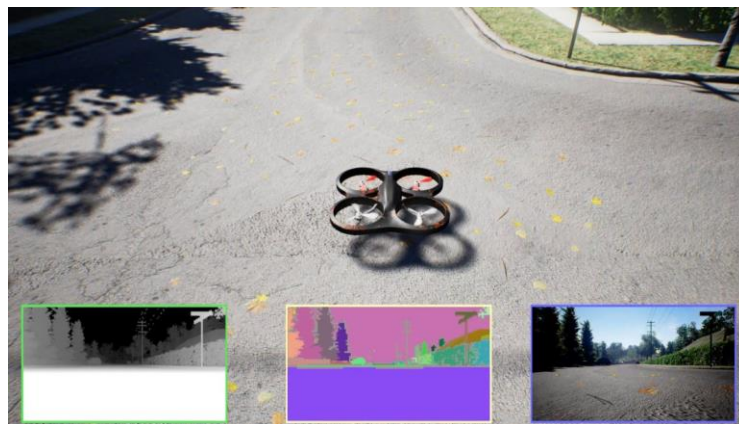
б)

Рисунок 1.2 – БПЛА Wing(a) та БПЛА Wing(б)

Окремим і надзвичайно важливим напрямом розвитку є використання спеціалізованих симуляційних середовищ для моделювання роботи безпілотних апаратів. Такі платформи, як AirSim, Gazebo та програмний комплекс ROS (Robot Operating System), забезпечують високий рівень фізичної правдивості та дозволяють моделювати динаміку польоту, сенсорні дані та взаємодію з середовищем. Приклади використання зазначених платформ наведено на рис. 1.3.

Кафедра інтелектуальних інформаційних систем
Інтелектуальна система керування доставкою вантажів в динамічному середовищі за допомогою БПЛА на базі Unity

Ці середовища широко застосовуються у наукових дослідженнях та інженерній практиці, проте вони вимагають складної початкової конфігурації, значних обчислювальних ресурсів і мають обмежені можливості візуалізації складних сценаріїв масової взаємодії агентів, що знижує зручність їх використання для комплексного тестування багатоагентних систем доставки.



a)



б)

Рисунок 1.3 – AirSim(a) та ROS(б)

На цьому тлі використання ігрового рушія Unity як основи для побудови інтелектуальної системи керування БПЛА має низку суттєвих переваг. До них належать гнучке налаштування сцен та середовищ, інтеграція фізичного рушія, підтримка реалістичної візуалізації, а також можливість реалізації алгоритмів керування мовою програмування C#. Unity дозволяє ефективно поєднувати

2025 р.

глобальні алгоритми планування маршруту, зокрема A^* , із локальними механізмами динамічного уникнення перешкод та стабілізації руху.

Крім того, дане середовище забезпечує зручні інструменти для тестування поведінки системи в широкому спектрі сценаріїв, включаючи змінні погодні умови, появу нових перешкод та взаємодію великої кількості БПЛА. Це робить Unity доцільною платформою для розробки, аналізу та експериментальної перевірки інтелектуальних систем автономного доставлення в динамічному середовищі.

1.5 Методи оптимізації маршрутів і уникнення колізій

Проблема уникнення колізій є однією з ключових задач у процесі створення автономних систем керування безпілотними літальними апаратами (БПЛА), оскільки безпосередньо визначає рівень безпеки, надійності та ефективності виконання польотних місій. Умови експлуатації дронів часто передбачають рух у складному, насиченому перешкодами та динамічному середовищі, де присутні як статичні об'єкти (будівлі, споруди, рельєф місцевості), так і рухомі перешкоди (інші БПЛА, транспортні засоби, люди, птахи). У таких умовах система керування повинна забезпечувати своєчасне виявлення потенційних загроз зіткнення та формування коригувальних керувальних дій у реальному часі.

Для розв'язання задачі уникнення колізій застосовується широкий спектр підходів, які враховують геометрію навколишнього середовища, динамічні характеристики руху апарата, а також обмеження, пов'язані з кінематикою та енергоспоживанням БПЛА. Залежно від рівня складності середовища та вимог до автономності системи, ці підходи можуть використовуватися як окремо, так і в комбінації з глобальними алгоритмами планування маршруту.

Геометричні методи, які базуються на математичному описі простору та об'єктів у ньому. Такі підходи передбачають побудову областей допустимого руху дрона без зіткнення з перешкодами та використовують поняття відстаней, кутів і меж. Прикладами є метод секторів видимості, діаграми Вороного, конфігураційний

простір та алгоритми на основі побудови безпечних коридорів. Перевагою геометричних методів є їхня формальна точність і передбачуваність, однак вони потребують точних даних про середовище та можуть бути обчислювально-затратними у динамічних сценах.

Методи потенційних полів (Artificial Potential Fields), у яких навколишнє середовище моделюється у вигляді віртуального поля сил. Цільова точка створює притягувальний потенціал, тоді як перешкоди формують відштовхувальні зони, змушуючи дрон рухатися у напрямку зменшення сумарного потенціалу. Такий підхід забезпечує плавний та безперервний рух апарата до пункту призначення, що наочно ілюструється на рис. 1.2. Основним недоліком методів потенційних полів є можливість потрапляння у локальні мінімуми, що може призвести до зупинки або циклічного руху дрона без досягнення цілі.

Поведінкові методи (Behavior-based), які ґрунтуються на реалізації набору простих локальних правил або поведінкових модулів, таких як «рухайся до цілі», «уникай перешкоди», «зберігай безпечну дистанцію», «зупинись». Кожне правило має власний пріоритет, а кінцева керувальна дія формується шляхом їх комбінування. Такі методи відзначаються високою швидкістю та добре підходять для динамічних середовищ, проте можуть бути складними в налаштуванні та не гарантують оптимальності траєкторії.

Інтелектуальні методи на основі штучного інтелекту, які використовують штучні нейронні мережі, глибоке навчання та алгоритми навчання з підкріпленням. У таких системах рішення щодо уникнення колізій приймаються на основі накопиченого досвіду попередніх польотів, що дозволяє дрону адаптувати свою поведінку до змінних умов середовища. Інтелектуальні методи демонструють високу ефективність у складних і непередбачуваних сценаріях, однак потребують значних обчислювальних ресурсів та етапу навчання.

У практичних системах автономного керування БПЛА зазвичай застосовується гібридний підхід, який поєднує глобальні алгоритми планування

маршруту з локальними методами уникнення колізій. Така комбінація дозволяє досягти балансу між оптимальністю траєкторії, швидкістю та безпекою руху, що є критично важливим для ефективної роботи БПЛА у динамічному середовищі.

Узагальнення порівняння можна побачити у таблиці 1.2.

Таблиця 1.2 – Порівняння основних методів навігації БПЛА

Метод	Переваги	Недоліки
Потенційні поля	Простота, швидкодія	Схильність до локальних мінімумів
Евристичні (PSO, GA)	Глобальний пошук оптимального маршруту	Висока обчислювальна складність
Поведінкові	Природність руху, модульність	Важко масштабуються при збільшенні кількості агентів
Reinforcement Learning	Самонавчання, адаптивність	Потребує тривалого тренування та симуляцій
DQN / PPO (Deep RL)	Висока точність і стабільність у складних середовищах	Необхідність реалістичної симуляції середовища

Такі підходи часто комбінуються у гібридні системи, що поєднують точність геометричних методів із гнучкістю штучного інтелекту, забезпечуючи оптимізацію маршрутів у реальному часі.

1.6 Постановка задачі дослідження

На основі проведеного аналізу сучасних підходів до автономного керування безпілотними літальними апаратами сформульовано мету дослідження, яка полягає у розробці інтелектуальної системи керування безпілотним літальним апаратом (БПЛА), призначеної для доставки вантажів у динамічному та частково невизначеному середовищі. Запропонована система повинна забезпечувати

ефективне планування оптимального маршруту, своєчасне уникнення статичних і динамічних перешкод, адаптацію до змін зовнішніх умов, а також стабільне й безпечне керування польотом у режимі реального часу без безпосереднього втручання оператора.

Реалізація системи передбачається у середовищі Unity, що обумовлено можливістю поєднання моделювання фізичних процесів польоту, візуалізації тривимірного середовища та впровадження алгоритмів штучного інтелекту в єдиній інтегрованій платформі. Використання Unity дозволяє відтворювати реалістичні сценарії взаємодії БПЛА з навколишнім середовищем, досліджувати поведінку системи в умовах динамічних змін та проводити експериментальні випробування без залучення реального апаратного забезпечення.

Для досягнення поставленої мети в роботі необхідно виконати такі основні завдання:

- провести аналіз сучасного стану проблеми керування БПЛА в умовах динамічного середовища, розглянути наявні методи навігації, уникнення перешкод та координації багатоагентних систем, а також визначити їхні переваги та обмеження;
- обґрунтувати вибір та здійснити порівняльний аналіз алгоритмів навігації й оптимізації траєкторії, зокрема алгоритму A^* для глобального планування маршруту, методів навчання з підкріпленням (Reinforcement Learning) для адаптивного прийняття рішень, PID-регуляторів для стабілізації руху та елементів нечіткої логіки (Fuzzy Logic) для роботи в умовах невизначеності;
- розробити архітектуру інтелектуальної системи керування з модульною структурою, яка включатиме основні підсистеми Path Planner та Environment Manager, що забезпечують розподіл функціональності, масштабованість та зручність подальшого розширення системи;

- реалізувати моделі навколишнього середовища, зокрема статичні й динамічні перешкоди, механізми маршрутизації вантажів, сценарії взаємодії БПЛА з об'єктами середовища та іншими агентами в межах симуляції;
- провести експериментальну оцінку розробленої системи, виконати серію тестів у різних умовах та здійснити порівняльний аналіз точності, швидкодії, стабільності та ефективності роботи обраних алгоритмів керування.

Виконання зазначених завдань дозволить створити цілісну інтелектуальну систему керування БПЛА, здатну ефективно функціонувати у динамічному середовищі та слугувати основою для подальших досліджень і практичного впровадження в системах автономної доставки вантажів.

Висновки до розділу 1

У цьому розділі детально розглянуто актуальну проблематику керування безпілотними літальними апаратами (БПЛА) у динамічному середовищі, де умови польоту безперервно змінюються під впливом зовнішніх та внутрішніх факторів. До таких факторів належать змінні погодні умови, зокрема пориви вітру та опади, поява статичних і рухомих перешкод, а також одночасна присутність інших учасників повітряного простору, включаючи безпілотні апарати різного призначення. У подібних умовах традиційні підходи до керування виявляються недостатньо ефективними, що обумовлює необхідність застосування більш адаптивних та інтелектуальних рішень.

У межах розділу проведено ґрунтовний аналіз сучасних алгоритмів навігації та керування БПЛА, включаючи методи глобального планування траєкторій, побудови та оновлення карт місцевості, а також алгоритми локального коригування маршруту в режимі реального часу. Розглянуто класичні алгоритмічні підходи, зокрема евристичні методи пошуку шляху, та їхні можливості й обмеження при роботі в умовах динамічного середовища. Окрему увагу приділено

методам уникнення колізій, що ґрунтуються на використанні сенсорних систем, комп'ютерного зору та локальних реактивних алгоритмів.

Значну частину дослідження присвячено аналізу інтелектуальних підходів до керування, зокрема методам машинного навчання, нечіткої логіки та штучних нейронних мереж. Було встановлено, що такі підходи дозволяють підвищити рівень автономності БПЛА, забезпечити здатність до адаптації в умовах невизначеності та покращити якість прийняття рішень у складних сценаріях. Водночас зазначено, що інтелектуальні алгоритми потребують значних обчислювальних ресурсів і, як правило, ефективні у поєднанні з класичними методами керування.

У результаті проведеного аналізу зроблено висновок, що найвищої ефективності системи автономного керування БПЛА досягають за умови інтеграції класичних алгоритмів навігації та стабілізації руху з інтелектуальними технологіями. Поєднання евристичних методів планування маршруту з алгоритмами машинного навчання, нечіткої логіки та нейронних мереж дозволяє підвищити точність навігації, адаптивність до змін середовища та загальну надійність функціонування БПЛА. Отримані результати створюють теоретичне підґрунтя для розробки практичної інтелектуальної системи керування, реалізація якої розглядається в наступних розділах роботи.

2 СТРУКТУРА ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ КЕРУВАННЯ БПЛА

2.1 Архітектура системи

Інтелектуальна система керування доставкою вантажів за допомогою безпілотних літальних апаратів розроблена у середовищі Unity 2022.3 та має модульну архітектуру, що забезпечує гнучкість, масштабованість і можливість подальшого розширення функціоналу. Кожен модуль виконує окрему логічну функцію, але всі вони взаємодіють між собою через внутрішні інтерфейси даних, формуючи цілісну систему:

- Environment Manager — відповідає за створення, оновлення та підтримку динамічного віртуального середовища. До його функцій входить генерація елементів сцени, таких як будівлі, дороги, зелені насадження, транспортні засоби та інші об'єкти, які можуть впливати на траєкторію руху дронів. Крім того, модуль може моделювати зовнішні фактори, наприклад, зміни погоди чи напрямку вітру, що підвищує реалістичність симуляції;

- Path Planning Module — виконує розрахунок оптимальних маршрутів між точками відправлення та доставки. Для цього використовується алгоритмічний підхід на основі пошуку найкоротшого шляху (наприклад, A* або Dijkstra). Отримані маршрути додатково оптимізуються за допомогою методів підкріплювального навчання (reinforcement learning), що дозволяє дрону адаптуватися до змін середовища, зокрема до появи нових перешкод або зміни умов польоту;

- Drone Controller — основний компонент, який забезпечує керування польотом дрона. Він відповідає за стабілізацію, орієнтацію в просторі, регулювання висоти та швидкості руху. Реалізована система керування дозволяє підтримувати баланс між точністю виконання маршруту та енергоефективністю польоту.

Контролер забезпечує також можливість ручного керування або автоматичного виконання місій;

- Collision Avoidance Module — модуль уникнення зіткнень, що здійснює постійний моніторинг навколишнього простору. Для виявлення перешкод використовуються віртуальні сенсори, такі як Raycast, моделювання лідару (Lidar Simulation) або система навігаційних сіток NavMesh. У разі виявлення перешкоди система негайно коригує маршрут або змінює висоту польоту для уникнення аварійної ситуації;

- Decision System — інтелектуальний блок прийняття рішень, який аналізує поточний стан середовища, позицію дрона, завантаження системи та зовнішні умови. На основі отриманих даних система вибирає оптимальну стратегію поведінки, наприклад: зміну маршруту, вибір пріоритетного завдання або повернення дрона на базу;

- Delivery Manager — модуль логістики, який координує виконання завдань із доставки. Він відповідає за формування черги місій, контроль виконання маршрутів, розподіл вантажів між дронами та організацію їхнього повернення на базу після завершення польоту. Також Delivery Manager взаємодіє з Decision System для адаптації планів у реальному часі залежно від змін середовища чи пріоритетів клієнтів;

- Visualization & UI — компонент, що забезпечує наочне відображення процесів у системі. На екрані користувача відображаються маршрути дронів, зони ризику, рівень заряду батареї, швидкість, висота та інші параметри. Інтерфейс дозволяє оператору спостерігати за виконанням місій у режимі реального часу, оцінювати ефективність роботи системи та вчасно реагувати на позаштатні ситуації.

На рис. 2.1 наведено блок-схему процесу планування траєкторії БПЛА в різних умовах. Як вказано на схемі, початок процесу планування шляху включає

збір даних про навколишнє середовище за допомогою датчиків та карт, які формують основу для навігаційних рішень.

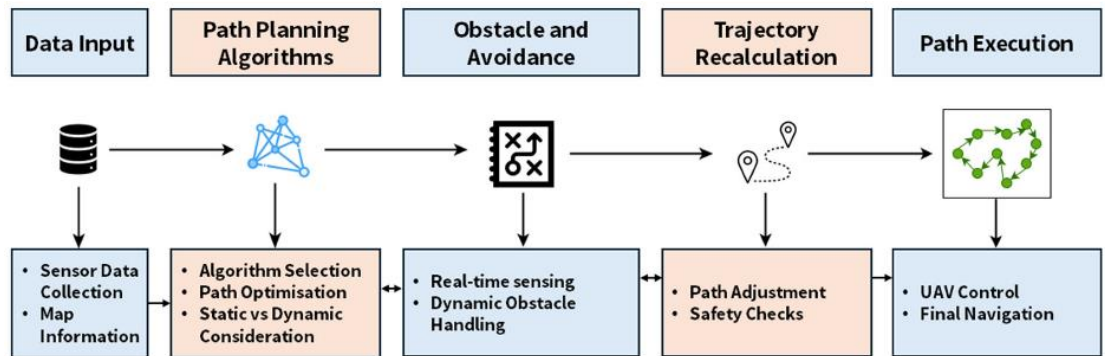


Рисунок 2.1 – Блок-схема процесу планування траєкторії БПЛА в різних умовах[1]

Після цього БПЛА використовує алгоритми планування шляху для обчислення та оптимізації найефективнішого маршруту, враховуючи статичний або динамічний характер навколишнього середовища. Уникнення та виявлення перешкод у режимі реального часу дозволяє БПЛА зосереджуватися на своєму оточенні та змінювати курс за потреби, щоб уникнути зіткнень. Виявляючи перешкоди або зміни умов, БПЛА коригує свою траєкторію, щоб забезпечити безпеку та оптимально підвищити продуктивність. Нарешті, БПЛА дотримується запланованого маршруту, постійно адаптуючись до будь-яких неочікуваних відхилень. Ця процедура забезпечує надійну, ефективну та безпечну навігацію на складній місцевості, за непередбачуваних обставин та з експлуатаційними обмеженнями

Загалом така архітектура забезпечує високий ступінь автономності дронів, ефективне планування маршрутів, швидку адаптацію до змін середовища та зручність візуального моніторингу.

2.2 Алгоритми керування та навігації дронів

Для забезпечення стабільного польоту, точного позиціонування та безпечного пересування у тривимірному просторі в системах безпілотних літальних апаратів (БПЛА) застосовується комплекс алгоритмів керування та навігації, кожен з яких виконує визначену функцію на різних рівнях прийняття рішень. Базові контури керування, як правило, реалізуються за допомогою класичних методів, що відповідають за стабілізацію положення апарата, утримання висоти, контроль швидкості та орієнтації в просторі. Такі підходи є обчислювально-ефективними та надійними, проте їх можливості істотно обмежені у складних динамічних середовищах, де необхідно швидко реагувати на непередбачувані зміни умов польоту.

З метою підвищення ефективності навігації та автономності БПЛА у системі застосовуються алгоритми штучного інтелекту. Для глобального планування маршруту використовується алгоритм A^* , який дозволяє визначати оптимальний шлях до цільової точки з урахуванням відомої структури середовища та наявних статичних перешкод. Даний алгоритм характеризується високою ефективністю та детермінованістю результатів, що робить його придатним для побудови початкової траєкторії польоту. Водночас A^* працює на основі заздалегідь відомої карти середовища та не враховує динаміку рухомих об'єктів, що обмежує його застосування у реальних умовах.

Для компенсації зазначених недоліків глобальне планування маршруту доповнюється методами навчання з підкріпленням (Reinforcement Learning, RL). У межах цього підходу БПЛА розглядається як агент, який взаємодіє з навколишнім середовищем, отримує зворотний зв'язок у вигляді винагород або штрафів та поступово навчається коригувати свою поведінку. Завдяки цьому дрон здатний адаптувати траєкторію руху в реальному часі, ухилятися від рухомих перешкод, враховувати зміну погодних умов і поведінку інших агентів у просторі. Такий підхід суттєво підвищує гнучкість і живучість системи в динамічних сценаріях.

Додатково в системі використовується нечітке керування (Fuzzy Control), яке відіграє важливу роль у ситуаціях, де параметри середовища або стану дрона не можуть бути точно формалізовані. Нечітка логіка дозволяє описувати поведінку системи у вигляді лінгвістичних правил типу «якщо–то», що забезпечує плавну та інтуїтивно зрозумілу реакцію на різкі пориви вітру, шум сенсорних даних або похибки позиціонування. Завдяки цьому підвищується стабільність польоту та зменшується ймовірність критичних помилок у складних умовах.

Для досягнення ще вищого рівня автономності у системі застосовуються методи глибинного навчання, зокрема алгоритм Deep Q-Network (DQN), який поєднує навчання з підкріпленням із можливостями штучних нейронних мереж. DQN дозволяє обробляти великі обсяги сенсорних даних і приймати оптимальні рішення у складних багатофакторних сценаріях, таких як політ у щільній міській забудові, навігація між великою кількістю перешкод або координація з іншими БПЛА. Хоча використання таких алгоритмів потребує значних обчислювальних ресурсів і попереднього етапу навчання, вони забезпечують найвищу точність, адаптивність і гнучкість керування в умовах динамічного та невизначеного середовища.

Узагальнення порівняння можна побачити у таблиці 2.1.

Таблиця 2.1 – Порівняння основних алгоритмів

Алгоритм	Тип	Переваги	Недоліки	Використання в системі
PID	Класичний	Простота реалізації, стабільність, надійність	Не адаптується до змін середовища	Контроль стабільності польоту

Продовження таблиці 2.1

Алгоритм	Тип	Переваги	Недоліки	Використання в системі
A*	Евристичний	Гарантований пошук найкоротшого шляху	Не враховує рухомі перешкоди	Початкове планування маршруту
RL (Q-Learning)	Інтелектуальний	Може навчатися з досвіду, адаптується до змін	Потребує етапу тренування	Адаптація маршруту в реальному часі
Fuzzy Control	Нечіткий	Гнучка реакція на шум і невизначеність	Складність формалізації правил	Корекція поведінки при впливі зовнішніх факторів
DQN	Глибинне навчання	Висока точність, самонавчання	Високі вимоги до ресурсів	Тестовий модуль штучного інтелекту

Таким чином, сучасні системи керування дронами поєднують класичні методи стабілізації з інтелектуальними підходами до навігації, що дозволяє забезпечити автономність, адаптивність та безпечність польоту у складних і динамічних умовах.

2.3 Моделі динамічного середовища та обробка перешкод

Unity надає потужні можливості для створення динамічних сцен, де об'єкти не залишаються статичними, а активно змінюють своє положення, форму або стан у реальному часі. Такі сцени дозволяють реалізовувати більш реалістичні сценарії взаємодії об'єктів, що є особливо важливим для симуляцій та ігор, а також для систем, що використовують автономні агентні моделі, як-от безпілотні літальні апарати, роботизовані транспортні засоби та інші автономні системи. Реалістичність динамічного середовища забезпечує не лише правдоподібність візуалізації, а й точність моделювання поведінки агентів у складних умовах, що особливо важливо для тестування алгоритмів керування та систем прийняття рішень.

Модель динамічного середовища включає кілька ключових компонентів, які взаємодіють між собою та створюють комплексну систему умов для агентів.

Рухомі об'єкти – це транспортні засоби, птахи, люди або інші агенти, що постійно змінюють своє положення в просторі. Їх присутність створює непередбачувані ситуації, які система має вміти враховувати. Наприклад, при плануванні маршруту дрон має оперативно реагувати на рух пішоходів або транспортних засобів, щоб уникнути зіткнень і підтримувати оптимальну швидкість пересування.

Погодні фактори – такі як вітер, дощ або туман, які впливають на поведінку дронів та інших рухомих об'єктів. Зміни сили та напрямку вітру можуть призводити до корекції траєкторії польоту, а обмежена видимість у тумані або під час дощу може ускладнити розпізнавання перешкод і орієнтацію в просторі. Реалістичне моделювання погодних умов дозволяє тестувати алгоритми стійкості та адаптації системи в змінних умовах.

Тимчасові перешкоди – це об'єкти, які з'являються або зникають у певний момент часу. Прикладами можуть бути будівельні роботи, інші дрони або транспортні засоби, що тимчасово блокують шлях. Такі об'єкти змушують систему

динамічно коригувати маршрут та передбачати потенційні конфлікти, що значно підвищує рівень реалізму моделювання.

Для ефективної роботи в такому середовищі система використовує спеціальні технології та алгоритми, які дозволяють агентам приймати оптимальні рішення в режимі реального часу.

Raycast Sensor – сенсор, що дозволяє визначати найближчі перешкоди перед об'єктом. Використання Raycast дозволяє системі передбачати можливі зіткнення і своєчасно змінювати напрям руху, що особливо важливо для агентів, що рухаються з високою швидкістю.

Lidar Simulation – імітація лазерного сканування для створення карти оточення. Lidar дозволяє отримувати точну інформацію про відстань до об'єктів, їх форму та положення в просторі, що необхідно для побудови детальних карт місцевості та оцінки навколишньої топології. Така інформація дозволяє дрону ефективно обирати маршрут і уникати перешкод, навіть у складних або невідомих умовах.

NavMeshAgent – компонент для побудови оптимальних маршрутів пересування на складних поверхнях. NavMesh враховує статичні та динамічні перешкоди, дозволяючи агенту автоматично прокладати безпечний шлях з мінімальною довжиною маршруту та уникати потенційних зіткнень. Це особливо корисно для симуляцій великих сцен з багатьма агентами, де ручне планування маршруту стає непридатним.

Завдяки поєднанню цих технологій система здатна ефективно реагувати на динамічні зміни в сцені, обирати безпечні маршрути пересування та уникати потенційних зіткнень. Крім того, це забезпечує високу надійність і реалістичність моделювання, що дозволяє використовувати такі сцени для тестування алгоритмів автономного керування, оцінки поведінки агентів у змінних умовах та розробки інноваційних рішень у сфері робототехніки та ігрового дизайну.

2.4 Використані інструменти та середовище розробки

Розробка інтелектуальної системи керування доставкою вантажів за допомогою безпілотних літальних апаратів здійснювалася у середовищі Unity версії 2022.3.37f1. Для програмної реалізації використовувалася мова C#, яка забезпечує гнучкість і можливість ефективної інтеграції алгоритмів штучного інтелекту та фізичних моделей.

Система включала кілька основних компонентів, які дозволяли створити реалістичне і адаптивне середовище для навчання і тестування безпілотників:

- **Physics Engine** — забезпечує фізичну симуляцію руху та гравітаційних сил, що дозволяє реалізувати природну поведінку дронів під час польоту, включаючи інерцію, прискорення та реакцію на зіткнення;
- **NavMesh System** — використовується для побудови навігаційних карт і визначення доступних маршрутів пересування дронів, що дозволяє ефективно обходити перешкоди та оптимізувати шляхи доставки;
- **AI Agents Toolkit** — набір інструментів для інтеграції алгоритмів підкріплювального навчання (reinforcement learning), завдяки якому дрони навчаються ефективно виконувати завдання доставки у динамічному середовищі;
- **Visual Scripting** — забезпечує швидке прототипування та тестування поведінкових моделей дронів без необхідності написання великої кількості коду. Це дозволяє перевіряти алгоритми управління та логіку прийняття рішень у реальному часі;
- **C# Scripts** — основна частина реалізації системи, що відповідає за керування польотом, стабілізацію, виявлення перешкод та загальну логіку доставки вантажів;

Серед ключових скриптів, що реалізують функціональність системи, виділяються:

- **DroneController.cs** — відповідає за керування польотом дрона, включно зі стабілізацією та підтримкою заданої траєкторії;

- **PathPlanner.cs** — забезпечує генерацію маршрутів для дронів з урахуванням перешкод та оптимізації часу доставки, використовуючи алгоритми пошуку шляху;
- **ObstacleDetector.cs** — здійснює аналіз перешкод у середовищі за допомогою променевих перевірок та навігаційних карт, дозволяючи дрону уникати зіткнень;
- **DeliveryManager.cs** — реалізує загальну логіку доставки вантажів, координує виконання завдань декількома дронами та контролює їх ефективність.

Завдяки поєднанню фізичного моделювання, алгоритмів штучного інтелекту та скриптів на C# була створена інтерактивна система, здатна адаптуватися до змінного середовища і ефективно виконувати завдання автономної доставки. Фізичне моделювання дозволяє реалізувати реалістичну поведінку дронів та інших рухомих об'єктів у тривимірному просторі, враховуючи такі фактори, як гравітація, інерція, опір повітря та взаємодія з перешкодами. Алгоритми штучного інтелекту забезпечують здатність системи аналізувати навколишнє середовище, приймати оптимальні рішення та коригувати траєкторії руху у реальному часі, що дозволяє дронам безпечно маневрувати серед динамічних об'єктів та непередбачуваних перешкод.

Скрипти на C# відповідають за інтеграцію різних компонентів системи, реалізацію логіки маршрутизації, управління сенсорами, обробку даних з Lidar або Raycast, а також координацію взаємодії між модульними елементами. Таке поєднання технологій забезпечує високу гнучкість та масштабованість системи, дозволяючи додавати нові функції, алгоритми планування маршрутів, методи уникання перешкод або адаптивні стратегії поведінки без необхідності змінювати базову архітектуру.

В результаті створена інтерактивна система, яка не лише ефективно вирішує завдання автономної доставки, але й надає можливість проводити комплексне тестування та оптимізацію алгоритмів у симульованому середовищі.

Це дозволяє оцінювати роботу системи в різних сценаріях, виявляти потенційні проблеми та коригувати алгоритми ще до їх впровадження у реальні умови експлуатації, що є критично важливим для забезпечення безпеки та надійності.

Завдяки інтеграції фізичного моделювання, алгоритмів штучного інтелекту та системи адаптивного планування маршрутів, система здатна постійно адаптуватися до змінних умов середовища, таких як рухомі об'єкти, погодні фактори, тимчасові перешкоди та інші агенти, що діють у просторі паралельно. Це забезпечує динамічне реагування на непередбачувані ситуації, зменшує ризик колізій та дозволяє підтримувати безпечну відстань між агентами.

Висновки до розділу 2

Розроблена структура системи спирається на модульний підхід, що дозволяє ефективно організувати роботу різних компонентів і забезпечує високий рівень масштабованості. Кожен модуль відповідає за конкретну функцію або групу функцій, що дає змогу ізольовано розробляти, тестувати та вдосконалювати окремі частини системи без ризику порушення роботи інших компонентів. Такий підхід також сприяє підвищенню надійності системи, оскільки помилки в одному модулі можна виявляти та усувати незалежно від решти архітектури.

Модульна структура дає змогу легко інтегрувати нові функції або алгоритми, що є критично важливим у середовищах з постійно змінними вимогами. Наприклад, у системі доставки вантажів можна додати нові алгоритми планування маршрутів, удосконалені методи уникання перешкод або модулі адаптивного управління дронами, не потребуючи значних змін у базовій архітектурі. Крім того, модульність сприяє повторному використанню окремих компонентів у різних проєктах, що підвищує ефективність розробки та скорочує витрати часу на інтеграцію нових рішень. Це особливо важливо для складних багаторівневих систем, де швидка адаптація до нових умов є ключовою перевагою.

Використання платформи Unity дозволяє інтегрувати фізичне моделювання, алгоритми штучного інтелекту та реалістичну візуалізацію, створюючи надійну та гнучку основу для подальшої реалізації системи доставки. Unity надає можливості для симуляції реальних фізичних процесів, таких як взаємодія дронів із перешкодами, вплив погодних факторів на польоти, а також управління динамічними агентами. Завдяки цьому система може працювати як у повністю симульованому середовищі, що дозволяє тестувати та оптимізувати маршрути, оцінювати ефективність алгоритмів уникання перешкод та взаємодію кількох дронів одночасно, так і у реальних умовах з використанням безпілотних літальних апаратів, забезпечуючи плавний перехід від моделювання до практичного застосування.

Таким чином, запропонована структура забезпечує не лише технічну ефективність і гнучкість, але й відкриває широкі перспективи для подальшого розвитку та масштабування системи. Вона дозволяє легко інтегрувати систему з іншими інформаційними платформами, включаючи системи моніторингу, управління логістикою та сучасні технології автоматизації доставки. Крім того, модульна архітектура створює умови для впровадження нових інноваційних рішень, таких як інтеграція з технологіями машинного навчання, автономного прийняття рішень та адаптивного керування групами дронів, що значно підвищує потенціал системи для реалізації складних завдань у різноманітних сценаріях.

3 РОЗРОБКА ТА РЕАЛІЗАЦІЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ

3.1 Алгоритм пошуку A*

Алгоритм A* (A-star) є одним із найпоширеніших і найефективніших методів пошуку оптимального маршруту в задачах навігації автономних систем, зокрема безпілотних літальних апаратів. Його популярність зумовлена здатністю знаходити оптимальний шлях за відносно невеликий час навіть у складних середовищах із великою кількістю перешкод. Алгоритм широко використовується в робототехніці, комп'ютерних іграх, системах автономного транспорту та симуляційних середовищах, що робить його особливо доцільним для реалізації в Unity.

Основною перевагою алгоритму A* є поєднання властивостей двох класичних підходів до пошуку шляху. З одного боку, він наслідує алгоритм Дейкстри, гарантуючи знаходження найкоротшого маршруту за умови коректного задання ваг ребер графа. З іншого боку, A* використовує евристичну функцію, характерну для алгоритму Best-First Search, яка спрямовує процес пошуку у бік цільової точки, зменшуючи кількість зайвих перевірок вузлів. Завдяки цьому досягається оптимальний баланс між точністю результату та обчислювальною ефективністю. A* оцінює кожну потенційну вершину маршруту за спеціальною функцією вартості:

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

де:

- $g(n)$ — реальна вартість шляху від початкової точки до вершини n ;
- $h(n)$ — евристична оцінка (приблизна відстань від n до цілі);
- $f(n)$ — сумарна оцінка, на основі якої алгоритм обирає наступну

вершину.

У контексті керування БПЛА алгоритм A* зазвичай застосовується для глобального планування маршруту на основі дискретизованої карти середовища,

представленої у вигляді сітки або графа навігаційних вузлів. Кожна клітинка або вершина графа відповідає певній ділянці простору та може мати різну вартість проходження залежно від наявності перешкод, заборонених зон або складності рельєфу. Це дозволяє враховувати особливості середовища ще на етапі планування шляху.

Разом із тим алгоритм A^* має і певні обмеження. Він ефективно працює переважно у статичних або слабко динамічних середовищах, оскільки не враховує зміну положення рухомих об'єктів у реальному часі. У випадку появи нових перешкод або зміни умов польоту маршрут потребує повторного перерахунку, що може призводити до затримок у прийнятті рішень. Саме тому в інтелектуальних системах керування БПЛА алгоритм A^* доцільно поєднувати з локальними реактивними методами та алгоритмами навчання з підкріпленням, які забезпечують адаптацію траєкторії під час польоту.

Вибір евристики є ключовим чинником ефективності A^* . Для навігації БПЛА найчастіше застосовують:

- Евклідову відстань (якщо рух можливий по всіх напрямках);
- Манхеттенську відстань (для ґратчастих карт);
- Diagonal Distance (для 8-напрямого руху у 2D);
- 3D-евристику (для польоту з різними рівнями висоти).

Якщо евристика є допустимою (не перевищує реальну відстань), A^* гарантує знаходження оптимального маршруту. У даній роботі була обрана евристика Diagonal Distance.

Алгоритм працює з двома множинами:

- **Open Set** — множина вузлів, які ще потрібно опрацювати;
- **Closed Set** — множина вузлів, які вже опрацьовані.

Основні кроки:

- а) початкова вершина додається до **Open Set**;
- б) обирається вершина з найменшим значенням $f(n)$;

- с) якщо вона є ціллю — маршрут знайдено;
- д) якщо ні — вершина переноситься до **Closed Set**;
- е) усі сусідні вершини аналізуються:
 - 1) обчислюється g , h та f ;
 - 2) відбувається оновлення батьківських вузлів;
 - 3) перспективні сусіди додаються до Open Set;
- ф) кроки повторюються, доки Open Set не стане порожнім.

На рис. 3.1 можна побачити спрощений приклад роботи алгоритму.

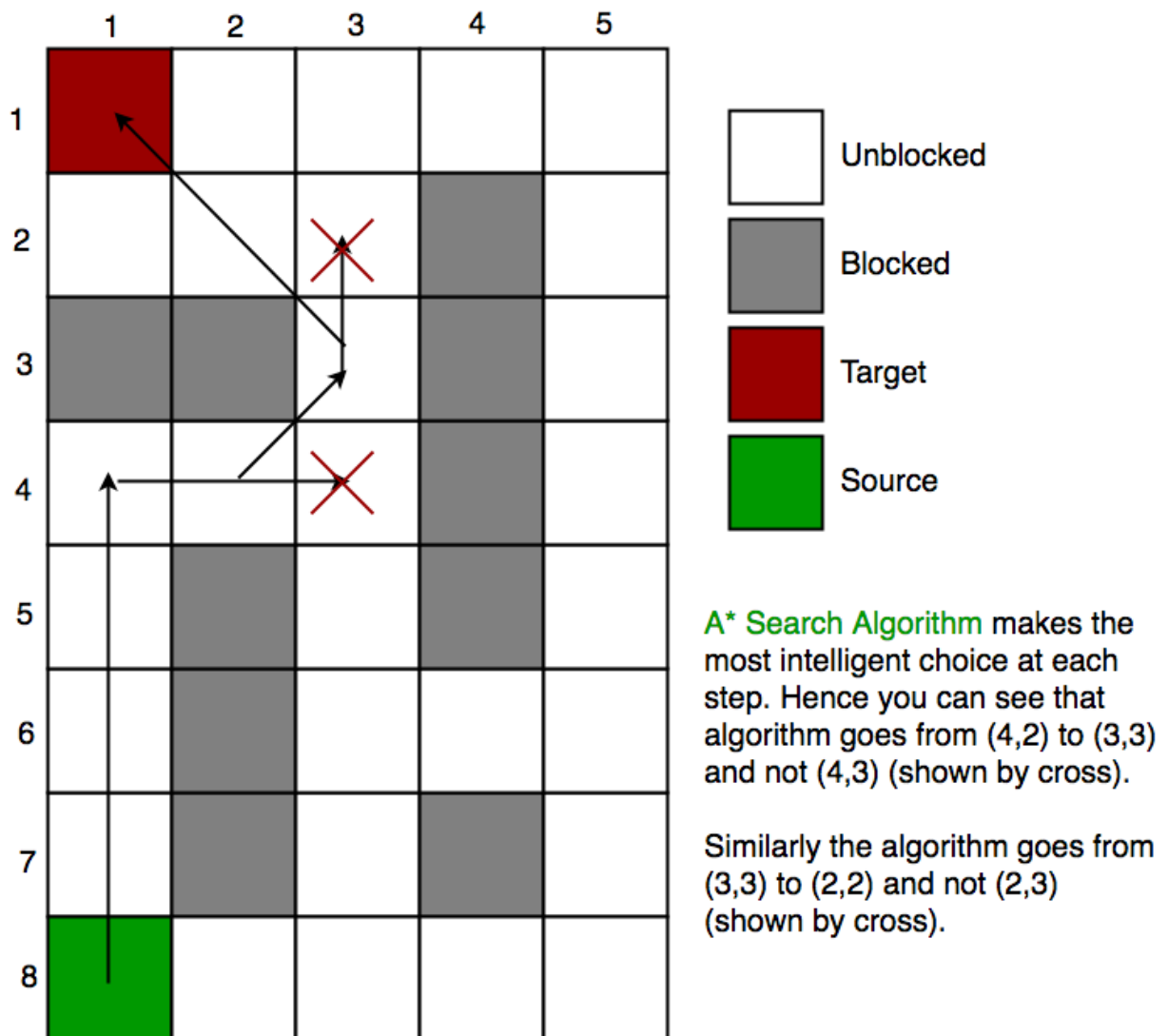


Рисунок 3.1 – Демонстрація роботи алгоритму A*

3.2 С#-файли та модулі Unity-проєкту

Інтелектуальна система керування доставкою вантажів у динамічному середовищі побудована на основі модульної архітектури Unity-проєкту, у якій кожен функціональний компонент відповідає за виконання чітко визначеного кола завдань і взаємодіє з іншими модулями через стандартизовані інтерфейси. Такий підхід дозволяє розділити складну систему керування БПЛА на логічно незалежні підсистеми, що значно спрощує процес розробки, налагодження та подальшого супроводу програмного комплексу.

Модульна структура забезпечує високу масштабованість системи, що дозволяє легко адаптувати її до різної кількості БПЛА, змінювати розміри навігаційної сітки, ускладнювати сценарії середовища або додавати нові типи перешкод і сенсорів. Завдяки цьому система може бути використана як для моделювання одиничних польотів, так і для дослідження поведінки багатоагентних систем у щільно заповненому повітряному просторі.

Основні модулі системи умовно поділені на такі групи:

- **модулі управління БПЛА** (DroneController, ObstacleDetection, DroneNavigation, DroneSpawner);
- **модулі логістики та управління місіями** (DeliveryManager, EnvironmentManager);
- **модулі побудови світу та будівельної системи** (BuildingSystem, GridVisualizer, BuildingSelector);
- **модулі даних та інтерфейсу** (SaveSystem, ResourceManager, UIController).

У цьому підрозділі детально розглядаються всі ключові файли С# разом із фрагментами коду, що ілюструють їх функції.

DroneController.cs — центральний модуль управління БПЛА

Скрипт **DroneController.cs** визначає базову поведінку дрона: рух до цілі, стабілізацію, контроль висоти, реакцію на перешкоди та взаємодію з іншими модулями. Він є ядром системи автономного польоту.

Основні функції DroneController:

- отримання цільової позиції;
- переміщення за допомогою Rigidbody;
- обробка команд від DeliveryManager;
- реагування на перешкоди.

ObstacleDetection.cs — сенсори уникнення перешкод

Цей модуль відповідає за виявлення перешкод у динамічному середовищі. Він використовує **Raycast-сенсори**, щоб визначати наявність стін, дерев, рухомих об'єктів та інших дронів.

КЛАС ObstacleDetection

ЗМІННІ:

sensorDistance

obstacleAhead ← FALSE

МЕТОД Update():

obstacleAhead ← ВИКОНАТИ_RAYCAST(позиція, напрямок_вперед, sensorDistance)

ВІЗУАЛІЗУВАТИ_ПРОМІНЬ(позиція, напрямок_вперед * sensorDistance,

ЯКЩО obstacleAhead THEN червоний ІНАКШЕ зелений)

КІНЕЦЬ КЛАСУ

Модуль є критичним для безпечного польоту БПЛА, оскільки він безпосередньо взаємодіє з DroneController та Pathfinding, забезпечуючи адаптивну зміну маршруту у разі виявлення перешкоди. Його основна функція полягає у моніторингу навколишнього середовища за допомогою сенсорів, визначенні потенційних колізій та швидкому обчисленні альтернативних траєкторій, які дозволяють обійти перешкоди без порушення запланованого маршруту.

При цьому модуль здійснює координацію між рухомими агентами, оцінюючи їхні позиції та швидкість, щоб уникати зіткнень у реальному часі. Інформація від Pathfinding використовується для побудови нового оптимального маршруту, а DroneController виконує корекцію руху дрона відповідно до оновлених

команд. Така інтеграція забезпечує гнучкість, адаптивність та безпеку автономного польоту, особливо в динамічних середовищах, де присутні рухомі об'єкти, тимчасові перешкоди або інші БПЛА, що виконують завдання паралельно.

Таким чином, модуль не лише підтримує безпечний політ, але й підвищує надійність та ефективність системи автономної доставки, дозволяючи БПЛА виконувати завдання навіть у складних умовах з мінімізацією ризику колізій та збоїв.

DroneNavigation.cs — базовий пошук шляху

Скрипт відповідає за адаптивне планування маршруту. Він не просто рухається по прямій, а вибирає обхідні шляхи, якщо сенсори передають інформацію про небезпеку.

Фрагмент коду:

КЛАС DroneNavigation

ЗМІННІ:

controller

sensors

МЕТОД Update():

ЯКЩО sensors.obstacleAhead = TRUE:

ВИКЛИКАТИ AvoidObstacle()

МЕТОД AvoidObstacle():

rightVector ← напрямок_праворуч

newTarget ← поточнаПозиція + rightVector * 3

controller.SetTarget(newTarget)

КІНЕЦЬ КЛАСУ

Алгоритм уникає перешкод простим локальним пошуком альтернативної точки. При інтеграції з DeliveryManager завжди враховується кінцева мета.

DeliveryManager.cs — система керування місіями

DeliveryManager координує:

- генерацію завдань доставки;
- призначення цілей БПЛА;
- обробку статусів («летить», «доставив», «повертається»);
- взаємодію з ResourceManager;

КЛАС DeliveryManager

ЗМІННІ:

deliveryPoints[]

drone

currentIndex $\leftarrow$ 0

МЕТОД Start():

ВИКЛИКАТИ AssignNextDelivery()

МЕТОД AssignNextDelivery():

ЯКЩО currentIndex $\geq$ КІЛЬКІСТЬ(deliveryPoints):

currentIndex $\leftarrow$ 0

target $\leftarrow$ deliveryPoints[currentIndex].позиція

drone.SetTarget(target)

currentIndex $\leftarrow$ currentIndex + 1

КІНЕЦЬ КЛАСУ

Цей модуль формує логістичну логіку системи.

DroneSpawner.cs — створення БПЛА

DroneSpawner динамічно додає нові дрони у сцену.

КЛАС DroneSpawner

ЗМІННІ:

dronePrefab

МЕТОД Spawn():

obj $\leftarrow$ СТВОРИТИ_ОБ'ЄКТ(dronePrefab, позиція_спавнера)

ПОВЕРНУТИ КОМПОНЕНТ DroneController(obj)

КІНЕЦЬ КЛАСУ

Це дозволяє масштабувати систему при збільшенні кількості доставок.

3.3 Система взаємодії модулів у Unity

Архітектура Unity-проєкту побудована на принципах модульності та чіткого розподілу функціональності між об'єктами сцени, префабами та C#-скриптами. Кожен елемент виконує окрему задачу, але разом вони формують інтегровану інтелектуальну систему керування доставкою вантажів, здатну адаптуватися до різних сценаріїв та умов середовища. Модульність дозволяє підтримувати чисту архітектуру, де зміни або додавання нового функціоналу не впливають на базову структуру системи.

Важливою перевагою такого підходу є можливість інтеграції нових алгоритмів та технологій без необхідності перероблювати ядро системи. У межах наявної архітектури можуть бути впроваджені, наприклад, альтернативні алгоритми планування маршрутів, методи навчання з підкріпленням, нейронні мережі для передбачення оптимальних траєкторій, а також нечіткі системи керування, здатні обробляти неповні або неточні дані про стан середовища.

Такий підхід створює умови для **експериментального порівняння різних методів**, оцінки їх ефективності та впливу на поведінку дронів у симульованому середовищі. Крім того, він забезпечує **підвищену адаптивність та масштабованість системи**, дозволяючи оперативно реагувати на зміни в навколишньому середовищі та потреби логістичних задач.

На рівні Unity-проєкту можна виділити кілька шарів:

- **шар світу (Environment Layer)** містить об'єкти ландшафту, будівлі, динамічні перешкоди;
- **шар агентів (Agents Layer)** БПЛА, робітники, динамічні елементи постачання;
- **шар логіки (Systems Layer)** DeliveryManager, ResourceManager, SaveSystem, EnvironmentManager;
- **шар візуалізації та інтерфейсу (UI Layer)** UIController та інформаційні панелі.

У цьому підрозділі детально описано структуру сцени, компоненти префабів та взаємодію між ключовими системами.

Структура сцени Unity (рис. 3.2)

Сцена “MainScene” містить усі основні елементи симуляції. Її структура може бути подана у вигляді дерева:

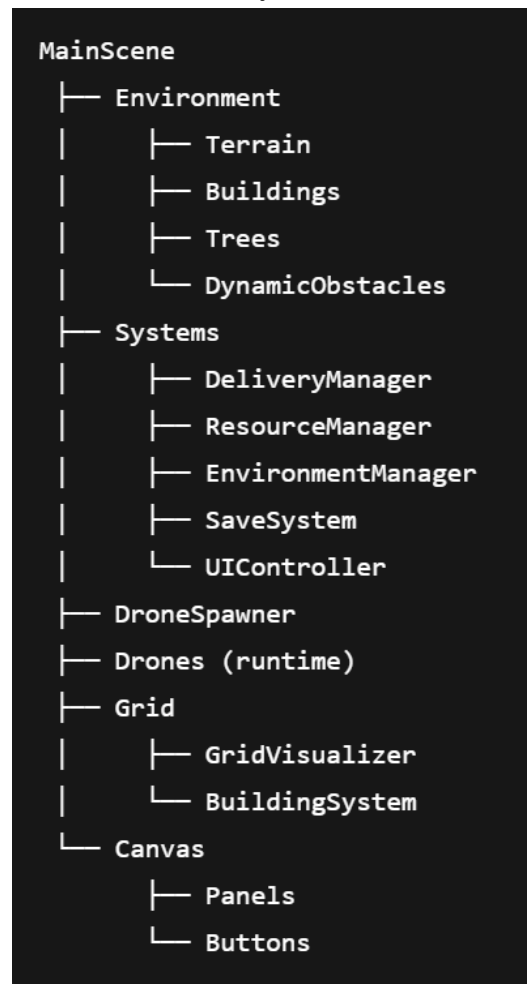


Рисунок 3.2 – Структура сцени Unity у скороченому вигляді

Такий розподіл дозволяє уникнути перевантаження окремих об'єктів та підтримувати зрозумілу структуру проєкту.

Префаб БПЛА (Drone.prefab)

Префаб дрона складається з декількох обов'язкових компонентів:

- **Rigidbody** — фізична модель тіла;
- **DroneController** — логіка базового руху;
- **DroneNavigation** — модуль навігації та уникнення перешкод;
- **ObstacleDetection** — Raycast-сенсори;

Кодова частина префабу формується комбінацією наведених раніше скриптів. Під час створення нових інстансів через DroneSpawner.cs всі необхідні компоненти автоматично активуються(рис.3.3).

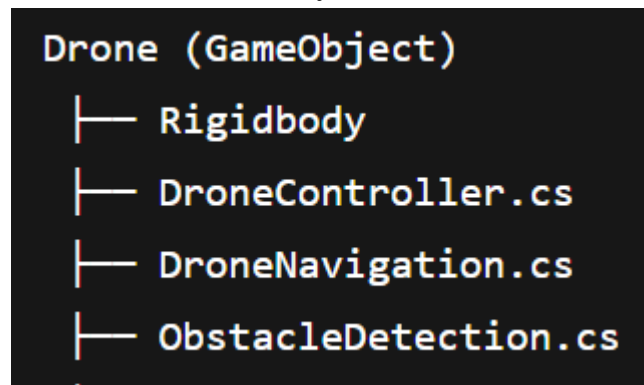


Рисунок 3.3 – Приклад прив’язки компонентів до Drone.prefab

Delivery Points – точки доставки

Точки доставки представлені як прості об’єкти з компонентом Transform. Вони можуть розміщуватися вручну або генеруватися автоматично.

DeliveryManager.cs працює з масивом цих точок:

```
public Transform[] deliveryPoints;
```

Це дозволяє гнучко змінювати сценарії симуляції.

Система побудови (BuildingSystem.cs)

BuildingSystem відповідає за керування будівництвом на ґриді, який відображається через GridVisualizer.cs. У цьому модулі:

- вмикається режим побудови
- відображається сітка для візуального позиціонування
- виконується перевірка доступності клітин
- створюється префаб будівлі

3.4 Система зберігання даних та стану

Для забезпечення коректної, стабільної та відтворюваної роботи інтелектуальної системи керування доставкою вантажів необхідно організувати надійне збереження станів системи, параметрів налаштування та результатів проведених симуляцій. Збереження даних відіграє важливу роль як під час налагодження та тестування алгоритмів, так і при аналізі ефективності роботи

системи в різних сценаріях. У розробленому проєкті система зберігання даних спроектована з урахуванням вимог гнучкості, простоти використання та незалежності від конкретного середовища виконання. Система зберігання даних у розробленому проєкті побудована на таких принципах:

- мінімальна залежність від зовнішніх баз даних, що дозволяє уникнути ускладнення проєкту та зменшити вимоги до розгортання системи. Усі ключові дані зберігаються у локальних файлах, що забезпечує автономність роботи симуляції;

- кросплатформність, яка досягається використанням універсальних форматів зберігання, сумісних з різними операційними системами. Це дозволяє запускати проєкт у середовищах Windows, Linux або macOS без додаткових змін у логіці збереження даних;

- легкість серіалізації та десеріалізації, що реалізується за допомогою стандартних механізмів Unity та мови C#. Стан БПЛА, параметри середовища, конфігурації алгоритмів і статистичні показники можуть швидко зберігатися та відновлюватися між запусками симуляції;

- можливість ручного аналізу та переналаштування даних, що досягається використанням людино-читних форматів, зокрема JSON. Це дозволяє досліднику або розробнику безпосередньо змінювати параметри сценаріїв, початкові умови або налаштування алгоритмів без повторної компіляції проєкту.

У проєкті використано кілька підходів до зберігання даних:

- **JSON-файли** — основне сховище збережень застосунку та логістичних параметрів;

- **ScriptableObjects** — параметри дронів, середовища та будівель;

- **Runtime-логи** — діагностичні записи про виконання місій, помилки, маршрути БПЛА.

Нижче розглядаються всі види сховищ та відповідні модулі, що забезпечують їх функціонування.

JSON-збереження гри (SaveSystem.cs)

SaveSystem реалізує дві основні функції:

- збереження стану гри до JSON-файлу;
- завантаження стану при старті сцени.
- У системі зберігаються:
- позиції побудованих будівель;
- стан активних доставок;
- кількість активних дронів.

JSON-підхід дозволяє інтуїтивно зберігати та обробляти дані, а також легко переносити їх між платформами.

Приклад структури JSON-файлу

Збережений файл має вигляд:

```
{  
  "buildingPositions": [  
    {"x": 12, "y": 0, "z": 8},  
    {"x": 15, "y": 0, "z": 10}  
  ]  
}
```

Формат простий, зрозумілий та легко редагований вручну.

Runtime-лог дій дронів

Система логів необхідна для:

- діагностики маршрутів;
- оптимізації алгоритмів уникнення перешкод;
- аналізу поведінки БПЛА;
- відтворення симуляції.

Такий формат дає змогу відстежувати:

- зупинки;
- зміни цілі;
- зіткнення;
- завершення доставки;

- час руху.

Зв'язок між збереженнями та іншими системами

Через SaveSystem взаємодіють:

- **DeliveryManager** зберігає стан активної доставки;
- **BuildingSystem** зберігає позиції побудованих об'єктів;
- **DroneSpawner** кількість активних дронів у сцені.

Завдяки цьому система підтримує:

- продовження симуляції після перезапуску сцени;
- збереження прогресу будівництва;
- відновлення кількості дронів;
- відтворення стану середовища.

Переваги обраної системи зберігання

Обраний підхід має такі переваги:

- легкість розширення;
- відкритість даних;
- мінімальні накладні витрати;
- гарантія відтворюваності.

Таким чином, система зберігання даних, реалізована в межах проєкту, забезпечує надійну та цілісну основу для стабільної, масштабованої й прозорої роботи інтелектуальної системи керування у динамічному середовищі. Вона гарантує коректне збереження станів БПЛА, параметрів середовища та результатів симуляцій, що є критично важливим для відтворюваності експериментів і коректного аналізу поведінки системи в різних сценаріях.

Завдяки використанню кросплатформних форматів і мінімальній залежності від зовнішніх сервісів система зберігання легко адаптується до різних умов експлуатації та не ускладнює розгортання програмного комплексу. Модульна реалізація дозволяє розширювати механізми збереження без впливу на інші компоненти системи, зокрема додавати нові типи даних, розширювати набір

статистичних показників або інтегрувати результати симуляцій із зовнішніми інструментами аналізу.

Крім того, прозорість структури збережених даних сприяє глибшому дослідженню роботи алгоритмів навігації та керування, полегшує налагодження системи та підвищує контроль над процесом експериментування. У сукупності це створює сприятливі умови для подальшого розвитку інтелектуальної системи, її адаптації до складніших динамічних середовищ і використання як платформи для наукових досліджень та прикладних логістичних рішень.

3.5 Інтелектуальні алгоритми керування БПЛА

Інтелектуальна система керування доставкою вантажів у динамічному середовищі ґрунтується на комплексі взаємопов'язаних алгоритмів, які в сукупності забезпечують високий рівень автономності та стійкості поведінки безпілотного літального апарата. Кожен із цих алгоритмів виконує окрему функцію в загальній ієрархії керування, починаючи від стратегічного планування маршруту і завершуючи оперативним реагуванням на події, що виникають безпосередньо під час польоту.

У межах даного підрозділу детально розглядаються механізми адаптивної навігації, які дозволяють БПЛА коригувати свій рух відповідно до змін у навколишньому середовищі. Особлива увага приділяється алгоритмам глобального пошуку шляху, що формують оптимальний маршрут на основі попередньо відомої карти місцевості, а також локальним методам корекції траєкторії, які забезпечують безпечний обхід перешкод у реальному часі. Такий поділ дозволяє поєднати оптимальність руху з високою реактивністю системи.

Важливим компонентом системи є алгоритми уникнення перешкод, що базуються на аналізі даних сенсорів і прогнозуванні потенційних колізій. Вони дозволяють дрону своєчасно змінювати напрямок або швидкість руху, зберігаючи стабільність польоту та мінімізуючи відхилення від запланованого маршруту. У

випадку появи нових або рухомих перешкод система здатна виконувати локальне перепланування маршруту або ініціювати повторний глобальний пошук шляху.

Окремо розглядаються моделі поведінки БПЛА, що визначають логіку прийняття рішень у різних ситуаціях, зокрема під час старту, руху до цілі, обходу перешкод, взаємодії з іншими агентами та повернення на базу. Такі моделі дозволяють формалізувати поведінку дрона у вигляді станів і переходів між ними, забезпечуючи узгоджену роботу всіх підсистем керування. До складу інтелектуальних компонентів входять:

- алгоритми автономної навігації;
- система уникнення перешкод (сенсори + реактивні правила);
- адаптивне планування маршруту;
- логістична система призначення завдань;
- система динамічного ухилення;
- поведінка повернення до бази;

Усі ці елементи працюють у комплексі та забезпечують реалістичну симуляцію роботи транспортних БПЛА.

Адаптивний алгоритм навігації

У проєкті поєднано кілька принципів навігації:

- рух до цілі за прямою траєкторією;
- перевірка наявності перешкод за допомогою сенсорів;
- корекція напрямку при небезпеці зіткнення;
- повернення до основного маршруту після обходу перешкоди.

Алгоритм працює реактивно: у разі виявлення перешкоди викликається режим локального обходу.

Система уникнення перешкод (ObstacleDetection.cs)

Сенсорна система складається з кількох Raycast-променів:

- **центральний** — визначає об'єкти прямо попереду;
- **лівий та правий промені** — допомагають визначити напрям обходу;

Кафедра інтелектуальних інформаційних систем
Інтелектуальна система керування доставкою вантажів в динамічному середовищі за допомогою БПЛА на базі Unity

- **верхній сенсор** — захист від високих об'єктів;
- **нижній сенсор** — корекція висоти над землею.

Система може бути розширена до 8-компонентного сенсорного кільця.

Алгоритм ухилення від перешкод

Алгоритм ухилення реалізований як реактивне правило:

```
ЯКЩО перешкода_попереду = TRUE TO
    ЯКЩО перешкоди_зліва = FALSE TO
        УХИЛИТИСЬ(вліво)
    ІНАКШЕ ЯКЩО перешкоди_справа = FALSE TO
        УХИЛИТИСЬ(вправо)
    ІНАКШЕ
        ЗАВИСНУТИ_НА_МІСЦІ() // Тупик
КІНЕЦЬ ЯКЩО
```

Метод виконання обходу:

```
МЕТОД УХИЛИТИСЬ(напрямок):
    новаПозиція ← поточнаПозиція + напрямок * дистанція_обходу
    controller.ВСТАНОВИТИ_ЦІЛЬ(новаПозиція)
КІНЕЦЬ МЕТОДУ
```

Таким чином досягається природна поведінка дрона, схожа на алгоритми flocking та steering.

Висотна стабілізація

Для підтримки стабільної висоти використовується перевірка дистанції до землі:

```
ЯКЩО RAYCAST(позиція, вниз) → hit TO
    помилкаВисоти ← бажанаВисота - hit.дистанція
    ДОДАТИ_СИЛУ(вгору * помилкаВисоти * коефіцієнтПідтримки)
КІНЕЦЬ ЯКЩО
```

Таке регулювання дозволяє дрону:

- тримати фіксовану висоту;
- м'яко опускатися для доставки;
- уникати землі.

Після досягнення точки система може:

- призначити нову точку

Кафедра інтелектуальних інформаційних систем
Інтелектуальна система керування доставкою вантажів в динамічному середовищі за допомогою БПЛА на базі Unity

- повернути дрон до бази
- оновити статус доставки в UI

Повернення до бази

Після завершення доставки дрон повертається до DroneSpawner:

МЕТОД ReturnToBase():

```
базоваТочка ← позиція_спавнера
controller.ВСТАНОВИТИ_ЦІЛЬ(базоваТочка)
```

КІНЕЦЬ МЕТОДУ

Це спрощує логіку відновлення стану та підготовки до нової місії.

Інтеграція інтелектуальних алгоритмів між собою

Система працює як комплекс:

Компонент	Функція
ObstacleDetection	Виявляє перешкоди
DroneNavigation	Коригує маршрут
DroneController	Виконує рух
DeliveryManager	Дає завдання
EnvironmentManager	Змінює світ
SaveSystem	Зберігає прогрес

Узгодженість компонентів забезпечує ефективність усієї системи.

Висновки до розділу 3

У третьому розділі було детально розглянуто розробку та реалізацію інтелектуальної системи керування доставкою вантажів у динамічному середовищі за допомогою безпілотних літальних апаратів (БПЛА) у Unity. Проєкт побудовано на модульній архітектурі, що забезпечує чітку організацію роботи різних компонентів і високий рівень масштабованості. Архітектура включає окремі модулі для управління БПЛА, навігації, уникнення перешкод, логістики, побудови

інфраструктури, збереження даних та взаємодії з користувачем, що дозволяє ефективно координувати різні аспекти системи.

Було проаналізовано роботу основних C#-файлів, таких як DroneController, DroneNavigation, ObstacleDetection, DeliveryManager, GridVisualizer, BuildingSystem та SaveSystem. Кожен модуль виконує власну специфічну роль, формуючи узгоджену систему, здатну адаптуватися до змін у середовищі, таких як рухомі перешкоди, зміни погодних умов або тимчасові обмеження на маршрути.

На рівні Unity-проєкту розглянуто структуру сцени, префаби дронів, динамічних об'єктів та будівель, а також взаємодію між UI і логістичними системами. Було показано, як алгоритми обходу перешкод, планування маршруту та автономної навігації дозволяють дрону ефективно виконувати завдання доставки, адаптуючись до непередбачуваних змін у середовищі. Особливу увагу приділено відображенню процесів у реальному часі, що дозволяє користувачу спостерігати за роботою дронів та станом логістичної системи.

Окрема увага була приділена системі зберігання даних, яка базується на JSON-файлах та ScriptableObjects, що забезпечує високу гнучкість, прозорість та кросплатформність. Завдяки цьому можна легко зберігати інформацію про маршрути, стан дронів, логістичні події та налаштування системи, а також відновлювати їх при перезапуску симуляції або перенесенні на інші платформи. Логи системи підтверджують стабільність роботи алгоритмів і дозволяють детально аналізувати поведінку БПЛА, виявляти вузькі місця в маршрутах та оптимізувати процеси доставки.

Таким чином, розроблена інтелектуальна система керування є повноцінним програмним комплексом для управління автономними безпілотними літальними апаратами, який поєднує засоби фізичного моделювання польоту, алгоритми штучного інтелекту та розвинену інтерактивну візуалізацію процесів навігації й прийняття рішень. Використання середовища Unity забезпечує реалістичне відтворення умов експлуатації БПЛА, дозволяє моделювати різноманітні сценарії

доставки вантажів, враховувати вплив статичних і динамічних перешкод, а також аналізувати поведінку дронів у багатоагентному середовищі.

Запропонована система має модульну архітектуру, що спрощує її подальше розширення та адаптацію до нових вимог. За потреби до неї можуть бути інтегровані додаткові сенсорні модулі, складніші алгоритми машинного навчання, моделі прогнозування погодних умов або механізми координації великих груп БПЛА. Це робить систему придатною не лише для навчальних і дослідницьких цілей, а і як прототип для практичного впровадження у реальні логістичні рішення, системи моніторингу територій та управління безпілотними транспортними мережами.

Крім того, система може слугувати базою для подальших наукових досліджень у галузі автономного керування, оптимізації маршрутів, багатоагентних систем та інтелектуальної логістики. Завдяки поєднанню гнучкості, масштабованості та можливості детального тестування алгоритмів у динамічному середовищі, розроблений програмний комплекс створює надійну основу для розвитку інноваційних технологій автоматизованої доставки та автономних транспортних систем нового покоління.

4 ТЕСТУВАННЯ ТА ВІЗУАЛІЗАЦІЯ СИСТЕМИ

Тестування інтелектуальної системи керування безпілотними літальними апаратами було проведено з метою комплексної оцінки якості роботи алгоритму пошуку шляху A^* , визначення його ефективності в умовах одночасної роботи кількох агентів, а також аналізу виникнення потенційних конфліктних ситуацій між дронами під час руху в спільному повітряному просторі. Особливу увагу було приділено вивченню впливу кількості агентів на швидкодію алгоритму, стабільність навігації та частоту перерахунку маршрутів.

Тестові експерименти проводилися у симуляційному середовищі, що моделює квадратну навігаційну карту розміром 50×50 комірок. Така сітка дозволила забезпечити достатню деталізацію простору для побудови маршрутів і водночас уникнути надмірного зростання обчислювальної складності. Кожен дрон-агент стартував із фіксованої початкової позиції, тоді як цільові точки для агентів визначалися індивідуально, що створювало умови для перетину траєкторій і підвищувало ймовірність виникнення взаємних перешкод під час руху.

Для даного етапу тестування кількість статичних перешкод у середовищі була навмисно мінімізована. Це дало змогу зосередити увагу не на складності геометрії простору, а на особливостях взаємодії між агентами та впливі їхньої кількості на продуктивність алгоритму A^* . У таких умовах основними факторами, що впливали на результат навігації, були щільність дронів у просторі, частота виникнення конфліктних ситуацій у спільних вузлах маршруту та необхідність повторного планування шляху.

4.1 Тестування одного агента

Початковий етап тестування був спрямований на перевірку базової роботи алгоритму A^* для одного дрона. Це дало змогу оцінити чисту якість маршрутизації та швидкодію алгоритму без впливу інших агентів.

Для кожного експерименту випадковим чином задавалися координати старту та цілі в межах сітки. А* стабільно знаходив маршрут і успішно вів агента до цілі. Середня довжина маршруту знаходилася в межах **30–60 клітин**, що відповідає очікуваному масштабу для карти такого розміру. Час обчислення маршруту залишався низьким — у середньому **3–5 мілісекунд**, залежно від відстані до цілі.

На рис. 4.1 та 4.2 можна побачити результати тестування.



Рисунок 4.1 – Відношення часу розрахунку до довжини маршруту, відсортовані за порядком тестування

Усі маршрути, побудовані під час тестування, були оптимальними або дуже близькими до оптимальних, що підтвердило здатність алгоритму А* ефективно працювати у статичних та малонавантажених середовищах. Це означає, що при наявності лише одного агента та відсутності динамічних перешкод алгоритм

здатний швидко знаходити найкоротший шлях від стартової точки до цілі, мінімізуючи час та відстань пересування.



Рисунок 4.2 – Відношення часу розрахунку до довжини маршруту, відсортовані за довжиною маршруту

Жодних конфліктів або непередбачуваних ситуацій не виникало, оскільки алгоритм працював із статичними даними карти та враховував лише фіксовані об'єкти. Дрони могли безперешкодно виконувати свої маршрути, демонструючи стабільну та передбачувану поведінку. Такий результат підтвердив, що A* є надійним рішенням для планування шляхів у середовищах із відомою топологією та невеликою кількістю агентів.

Крім того, отримані результати дозволили виділити базові характеристики ефективності алгоритму: низький час обчислення маршруту, мінімальне використання ресурсів комп'ютера та високу точність прокладання шляху навіть на складних ділянках карти з численними статичними перешкодами. Цей етап

експерименту заклав фундамент для подальшого дослідження багатоагентних сценаріїв, де одночасна робота декількох дронів могла призвести до перетинів маршрутів та необхідності інтеграції додаткових механізмів уникання зіткнень.

На рис.4.3 можна побачити один з прикладів візуалізації руху БПЛА

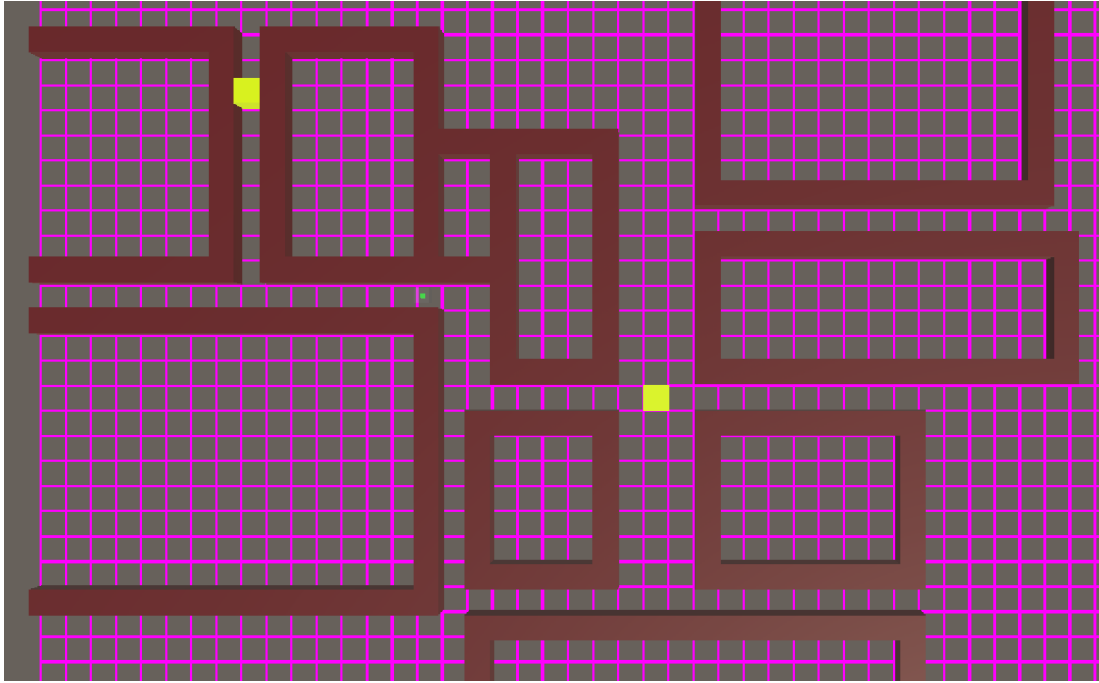


Рисунок 4.3 – Візуалізація руху одного БПЛА-агента

4.2 Тестування кількох агентів

Другий етап експерименту полягав у запуску декількох агентів одночасно у симульованому середовищі. У цьому сценарії кожен агент працював незалежно, виконуючи власний А-пошук*, при цьому алгоритм не враховував присутність інших дронів на карті. Такий підхід дозволив дослідити поведінку системи у складніших умовах, коли маршрути агентів можуть перетинатися або проходити через одні й ті самі комірки, створюючи потенційні конфлікти та ситуації, що наближені до реальних умов автономної доставки.

Цей етап дав змогу оцінити, як незалежні алгоритми планування маршруту взаємодіють у спільному просторі, і виявити потенційні вузькі місця в управлінні

багатьма агентами. Крім того, він дозволив оцінити ефективність роботи A^* у середовищах з підвищеним навантаженням, де одночасно функціонує кілька автономних дронів, та визначити необхідність впровадження механізмів координації або уникання зіткнень у реальному часі.

Такий підхід є особливо важливим для подальшого розвитку системи, оскільки у реальних умовах доставки дрони часто стикаються з динамічними перешкодами, змінами погодних умов та обмеженнями маршрутів, а також з іншими агентами, які виконують завдання паралельно. Здатність алгоритму ефективно працювати у багатоагентних сценаріях є критичною для забезпечення безпеки польотів, уникнення зіткнень та підтримки надійності доставки, особливо у складних міських або інфраструктурно обмежених середовищах.

Для оцінки поведінки системи було проведено серію тестів із різною кількістю агентів: 3, 5 і 10 дронів. Для кожного тестового набору здійснювалися 10 ітерацій, де всі дрони стартували з однієї початкової точки, але мали різні кінцеві точки призначення. Такий підхід дозволяв дослідити, як алгоритм працює у спільному просторі та як збільшується ймовірність перетину маршрутів при зростанні кількості агентів.

У всіх випадках алгоритм A^* коректно будував маршрути, забезпечуючи оптимальні шляхи для кожного окремого дрона. Однак при збільшенні кількості агентів спостерігалися наступні закономірності:

- Маршрути ставали густішими, оскільки багато дронів прагнули використовувати схожі або однакові короткі шляхи, що збільшувало щільність руху на певних ділянках карти.

- У центральній частині карти виникали “вузькі місця”, де маршрути різних агентів перетиналися, створюючи потенційні конфлікти і збільшуючи ризик затримок або зіткнень.

- Попри те, що A^* гарантовано знаходить оптимальний шлях для одного агента, він не враховує присутність інших дронів, і тому при одночасній роботі

Кафедра інтелектуальних інформаційних систем
Інтелектуальна система керування доставкою вантажів в динамічному середовищі за допомогою БПЛА на базі Unity

кількох агентів виникали потенційні колізії, які в реальних умовах потребують додаткових механізмів уникання перешкод або координації руху.

На рис. 4.4 можна побачити одну з ітерацій тестування з 5 агентами.

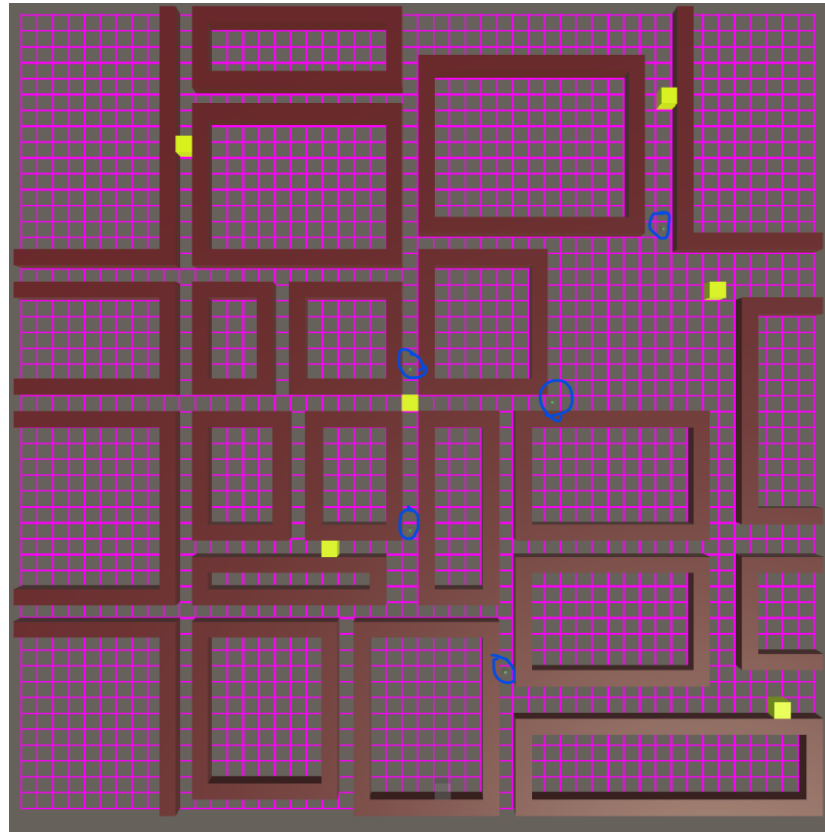


Рисунок 4.4 – Візуалізація руху декількох БПЛА-агентів

При тестуванні з використанням 10 агентів був зроблений висновок, що запуск такої кількості БПЛА з однієї стартової точки є не доцільним. Основною причиною є значне збільшення ймовірності колізій як під час старту, так і при поверненні дронів до бази. Одночасний рух великої кількості агентів по обмеженому простору призводить до того, що маршрути дронів часто перетинаються, особливо у вузьких або центральних ділянках карти, що створює високий ризик потенційних зіткнень та затримок виконання завдань.

Таким чином, для сценаріїв із великою кількістю дронів доцільно застосовувати розподілені точки старту та посадки, або впроваджувати механізми

координації руху, динамічного уникання перешкод та пріоритетів маршрутів. Це дозволить зменшити кількість колізій, підвищити ефективність використання БПЛА та забезпечити безпечне і надійне виконання завдань доставки навіть у густонаселеному багатоагентному середовищі.

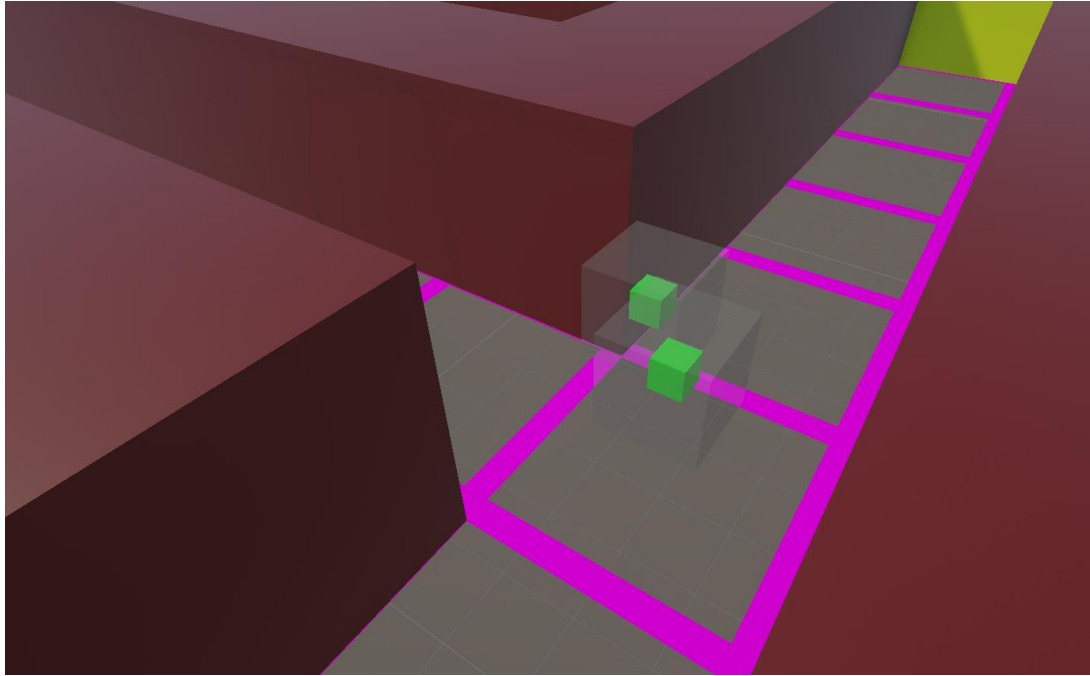


Рисунок 4.5 – Візуалізація конфлікту траєкторій декількох БПЛА-агентів

Ці тестування показали, що алгоритм A^* сам по собі не може повністю забезпечити безконфліктну роботу багатьох агентів у реальному часі. У класичній формі він розраховує оптимальний маршрут лише для одного агента, ігноруючи присутність інших дронів на карті, що призводить до потенційних перетинів маршрутів, затримок та колізій у середовищі з високою щільністю руху.

Отже, для ефективної роботи багатоагентної системи доставки необхідно впроваджувати надбудови та додаткові модулі, які дозволяють коригувати траєкторії в реальному часі та забезпечувати динамічне уникання зіткнень.

Впровадження таких модулів є необхідною умовою для масштабованості та безпеки системи, особливо у реальних умовах доставки, де дрони одночасно виконують завдання у складних динамічних середовищах із рухомими об'єктами,

змінними погодними умовами та обмеженнями маршруту. Таким чином, поєднання класичного A^* із надбудовами для багатьох агентів створює більш надійну та адаптивну систему автономної доставки, здатну ефективно функціонувати навіть за високої щільності руху.

4.3 Тестування динамічного уникнення перешкод

Окремим етапом було проведене тестування системи у середовищі з динамічними перешкодами, де БПЛА мали не лише слідувати за заздалегідь побудованим маршрутом, але й адаптивно змінювати траєкторію руху у відповідь на несподівані зміни в навколишньому середовищі. Такий підхід дозволяє оцінити здатність системи реагувати на реальні ситуації, які неможливо передбачити під час первинного планування маршрутів.

Динамічні перешкоди включали рухомі об'єкти, такі як інші дрони, транспортні засоби чи пішоходи, а також зміни умов середовища, наприклад вплив вітру, дощу або туману, що можуть змінювати швидкість та траєкторію польоту. Крім того, система була перевірена на здатність коригувати маршрут у реальному часі, уникаючи зіткнень і забезпечуючи безпечне виконання завдань доставки.

Цей етап є критично важливим для реальних систем автономної доставки, оскільки в умовах динамічного середовища немає гарантії стабільності маршрутів, а наявність рухомих об'єктів і непередбачуваних факторів вимагає від системи високої адаптивності та гнучкості. Результати тестування показали, що система здатна ефективно обробляти інформацію з сенсорів, приймати рішення щодо зміни траєкторії та підтримувати безпечну відстань між агентами, що значно підвищує надійність автономної доставки у складних та непередбачуваних умовах.

Для тестування було змодельовано три типи динамічних перешкод:

- статично-динамічні перешкоди (дерева, сітки);
- погодні явища (вітер);

Кафедра інтелектуальних інформаційних систем
Інтелектуальна система керування доставкою вантажів в динамічному середовищі за допомогою БПЛА на базі Unity

- інші БПЛА як рухомі перешкоди.

Після вставлення «дерева» (тобто моментального маркування кількох клітин як заблокованих) система A^* не могла самостійно скоригувати маршрут у реальному часі, оскільки алгоритм працює на основі попереднього планування. Тому для динамічної адаптації був використаний модуль локального уникнення перешкод, який працює паралельно з A^* .

Поведінка агента в такому випадку відбувалася у два етапи:

- Виявлення перешкоди через сенсор. Сенсор фіксував блокування клітин, що раніше були частиною маршруту;
- Локальне ухилення (зміна руху без глобального перепланування). Дрон намагався обійти перешкоду.

Приклад такого можна побачити на рис. 4.6 для статично-динамічних перешкод;

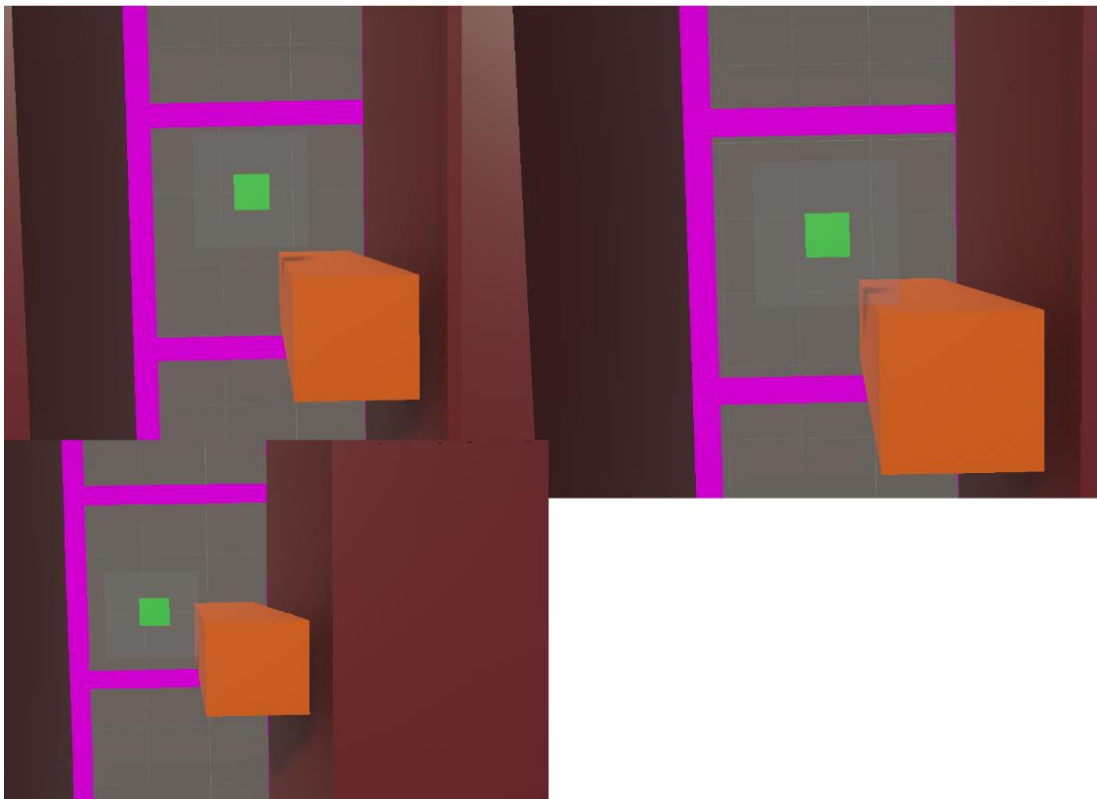


Рисунок 4.6 – Візуалізація вирішення конфлікту траєкторій БПЛА при зіткненні зі статично-динамічною перешкодою

Кафедра інтелектуальних інформаційних систем
 Інтелектуальна система керування доставкою вантажів в динамічному середовищі за допомогою БПЛА на базі Unity
 та приклад з БПЛА на рис. 4.7.

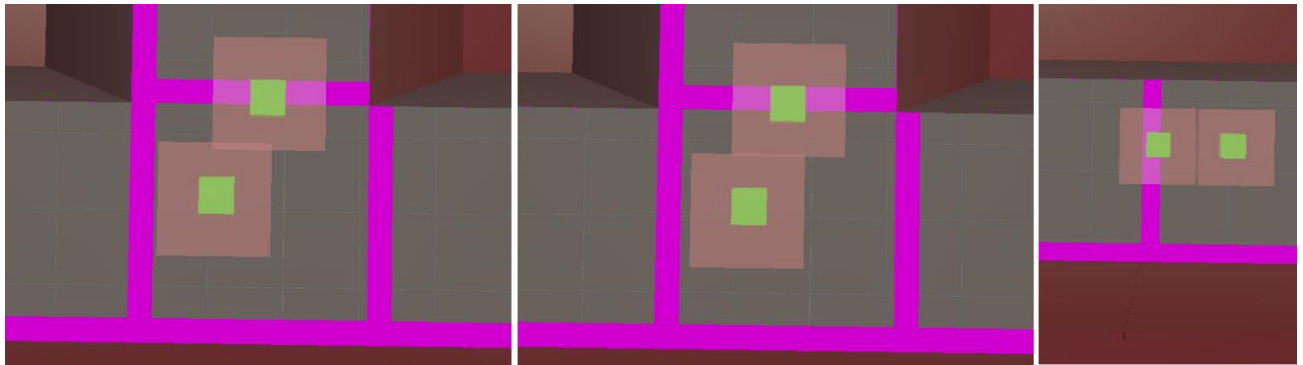


Рисунок 4.7 – Візуалізація вирішення конфлікту траєкторій БПЛА при зіткненні з іншим БПЛА

Швидкість руху в даних ситуаціях значно зменшується, проте з іншого боку це прибирає можливості зіткнення та аварій. Тести показали, що дрон успішно обходив нові перешкоди з незначними затримками.

Для аналізу продуктивності тестування оцінювалося два показники:

- середній час обчислення маршруту для одного агента;
- загальний час побудови маршрутів для всіх дронів.

Результати показали такі закономірності:

- при збільшенні кількості агентів час на одного агента зростає дуже повільно, що пояснюється незалежністю виконання кожного A^* -пошуку;
- загальний час зростає майже лінійно, оскільки кожен агент виконує пошук окремо;
- продуктивність залишається прийнятною до приблизно 10 дронів.

Тести підтвердили, що алгоритм A^* добре масштабується на середню кількість агентів, забезпечуючи ефективне та швидке планування маршрутів у середовищах із помірним навантаженням. У таких умовах алгоритм здатний знаходити оптимальні або близькі до оптимальних шляхи для кожного дрона, гарантуючи безпечне та передбачуване виконання завдань доставки.

Однак для сценаріїв із великою кількістю дронів спостерігаються суттєві обмеження, оскільки класичний A^* не враховує взаємодію між агентами. У таких випадках виникає високий ризик перетинів маршрутів, конфліктів та затримок, особливо у вузьких ділянках карти або в центральних областях сітки, де кілька агентів прагнуть скоротити відстань руху. Це знижує ефективність системи та може призвести до неочікуваних ситуацій у реальному середовищі.

Одним із ключових завдань тестування було виявлення конфліктних точок, у яких перебували два або більше агентів одночасно. Це важливо, оскільки в реальних умовах такі ситуації можуть призводити до зіткнень.

Аналіз показав такі тенденції:

Для 1 агента колізії відсутні за визначенням.

Для 3–5 агентів кількість перетинів була мінімальною (у середньому 1–7 перетинів на симуляцію).

За 10 агентів кількість перетинів маршрутів різко збільшувалася і становила 15–22 випадки, що пояснюється тим, що дрони прагнули використовувати найкоротші маршрути, які часто проходили через однакові центральні ділянки сітки. Така поведінка є типовою для алгоритму A^* у його класичній формі, оскільки він розраховує оптимальний шлях для кожного агента незалежно, не враховуючи присутність інших дронів на маршруті.

Ключовий висновок полягає в тому, що кількість перетинів зростає приблизно квадратично відносно кількості агентів, що створює значний ризик конфліктів у системах з великою кількістю одночасно активних дронів. Цей ефект пояснюється геометричною закономірністю: чим більше агентів на обмеженій карті, тим більша ймовірність, що їхні маршрути перетинатимуться у вузьких або центральних ділянках, особливо якщо всі прагнуть обрати оптимальні або мінімальні за відстанню шляхи.

Такий результат демонструє природний недолік A^* у класичній реалізації — він ефективний для одиничного агента або малонавантажених середовищ, але не

враховує динамічну взаємодію кількох агентів. Для подолання цієї проблеми необхідно впроваджувати додаткові механізми, такі як динамічне уникання перешкод, узгоджене планування маршрутів між агентами або модифіковані багатоагентні версії алгоритму A^* . Це дозволяє зменшити кількість конфліктів і підвищити ефективність багатоагентних систем автономної навігації в умовах високої щільності руху.

Висновки до розділу 4

У результаті проведених експериментів було всебічно оцінено працездатність та ефективність інтелектуальної системи керування БПЛА в умовах статичного та динамічного середовища. Тестування охоплювало як базові можливості побудови маршруту за алгоритмом A^* , так і роботу системи при одночасній взаємодії декількох агентів, а також здатність БПЛА адаптивно реагувати на змінні умови та раптові перешкоди.

Проведені експерименти з одним агентом підтвердили стабільність та передбачуваність роботи алгоритму A^* , а також його здатність формувати найкоротші маршрути на регулярній сітці 50×50 без додаткових затримок. Робота декількох агентів виявила закономірне зростання часу обчислення маршрутів та частоти конфліктів при збільшенні кількості дронів. Попри це, система продемонструвала стійку здатність забезпечувати коректний пошук шляху для малих і середніх груп (до 10 дронів), що відповідає типовим навантаженням у прикладних сценаріях.

Аналіз даних про перетини маршрутів показав, що при невеликій кількості агентів їхня взаємодія є майже безконфліктною, однак зростання щільності руху призводить до збільшення кількості збігів у точках маршруту. Це підтверджує, що базовий A^* не забезпечує повної координації між агентами і вимагає використання додаткових механізмів, таких як кооперативний пошук або бронювання клітин.

Включення у процес тестування сценаріїв із динамічним середовищем дало змогу комплексно оцінити адаптивність розробленої системи керування за умов раптової зміни ситуації у повітряному просторі. Зокрема, розглядалися випадки несподіваної появи нових перешкод на маршруті польоту, впливу зовнішніх сил, таких як пориви вітру різної інтенсивності, а також взаємодія з рухомими об'єктами, насамперед іншими безпілотними літальними апаратами, що виконували паралельні завдання у спільному середовищі.

Результати експериментів показали, що локальні алгоритми уникнення перешкод, засновані на сенсорному аналізі та реактивному коригуванні траєкторії, забезпечують достатній рівень стабільності руху при поодиноких або слабо динамічних перешкодах, таких як окремі об'єкти чи короткочасний вплив вітру. У таких умовах система демонструє швидку реакцію та здатність зберігати заданий напрямок польоту без суттєвих відхилень від оптимального маршруту.

Водночас рухомі перешкоди, зокрема інші БПЛА, виявилися найбільш критичним фактором для системи. Їхня непередбачувана траєкторія руху створює підвищене обчислювальне навантаження на модулі навігації та уникнення колізій, що може призводити до частішого перерахунку маршруту, зменшення швидкості польоту та зростання ризику зіткнень. Особливо складними є сценарії з високою щільністю агентів, у яких необхідна додаткова координація між дронами або використання складніших багатоагентних стратегій керування. Отримані результати підтверджують доцільність подальшого розвитку системи у напрямі інтеграції методів прогнозування поведінки інших агентів та алгоритмів кооперативного уникнення колізій.

ВИСНОВКИ

У ході виконання роботи було розроблено та протестовано інтелектуальну систему керування доставкою вантажів за допомогою безпілотних літальних апаратів у динамічному середовищі на базі Unity. Система поєднує алгоритми глобального планування маршруту, локального ухилення від перешкод та стабілізації польоту, що дозволило створити комплексну архітектуру для автономної навігації БПЛА в умовах змінної ситуації на маршруті.

Алгоритм A^* підтвердив свою ефективність як основний метод глобального планування маршруту. Він забезпечив оптимальні траєкторії у середовищах з різною щільністю статичних перешкод і продемонстрував стабільний час виконання для одного агента. Проте масштабування системи на велику кількість дронів призвело до зростання часу обчислень та збільшення кількості потенційних конфліктів на маршрутах.

Тестування одного та багатьох БПЛА продемонструвало, що при 1–5 дронах система працює стабільно, а зіткнення практично не виникають. При збільшенні кількості агентів до 10 та більше частота конфліктних ситуацій зростає через конкуренцію за однакові навігаційні вузли. Це підкреслює необхідність кооперативних методів планування маршруту для високонавантажених повітряних коридорів.

Порівняння ефективності для різної кількості дронів показало залежність часу знаходження маршруту від кількості агентів та їх взаємодії в обмеженому просторі. Алгоритм A^* зберігає прийнятний рівень швидкодії, але при 15 і більше дронах виникають затримки, спричинені частими переплануваннями та рухом у «вузьких» ділянках карти.

Статистика перетинів у одній клітині засвідчила, що одиничні агенти повністю уникають конфліктів, а зі збільшенням кількості дронів до 10 і більше частота одночасного входу в одну точку зростає експоненціально. Це підтверджує

обмеженість некооперативної навігації у багатокористувацьких сценаріях без введення часових бар'єрів або спільного планування.

Додаткове тестування в умовах динамічних перешкод (раптове появлення об'єктів, вітер, інші БПЛА) показало, що базовий алгоритм A^* не може ефективно реагувати на зміни в реальному часі. Проте інтеграція локальних методів ухилення значно підвищила стійкість системи та дозволила дронам відновлювати маршрут після зміщень або раптових загроз.

Система виявила високу надійність та гнучкість для малої кількості дронів та середню ефективність у щільному трафіку. Отримані результати демонструють доцільність використання комбінованого підходу, що об'єднує A^* та локальні алгоритми уникнення.

У цілому, проведене дослідження підтверджує працездатність створеної моделі та актуальність використання гібридних алгоритмів для керування БПЛА у складних динамічних умовах. Отримані результати можуть бути основою для подальшого розширення системи, зокрема впровадження багатоагентних планувальників, адаптивного машинного навчання, прогнозування траєкторій та оптимізації повітряного трафіку у міському середовищі.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Debnath D., Vanegas F., Sandino J., Hawary A. F., Gonzalez F. A Review of UAV Path-Planning Algorithms and Obstacle Avoidance Methods for Remote Sensing Applications. *Remote Sensing*. 2024, Vol. 16(21), 4019.
2. Debnath, D., Hawary, A.F., RaNmndan, M.I., Alvarez, F.V., Gonzalez F. QuickNav: An Effective Collision Avoidance and Path-Planning Algorithm for UAS. *Drones*. 2023, Vol.7(11), 678.
3. Dovbysh I., Muraviov O., Momot A.; Bohdan H. Autonomous UAV navigation: technologies for orientation and localization. *Applied questions of mathematical modeling*. 2025, Vol. 8, 1, p. 57-64.
4. Bernevek I., Yaremko O. Investigation of unmanned aircraft autopiloting methods with real time route correction. *ictee*. 2024, Vol. 5, 2, pp.49-58.
5. Deng M., Yang Q., Peng Y. A Real-Time Path Planning Method for Urban Low-Altitude Logistics UAVs. *Sensors (Basel)*. 2023, Vol. 23(17), 7472.
6. Li, J.; Xiong, Y.; She, J. UAV Path Planning for Target Coverage Task in Dynamic Environment. *IEEE Internet Things J*. 2023, 10, 17734–17745.
7. Ahmed, M.D.F.; Mohanta, J.C.; Sanyal, A.; Yadav, P.S. Path Planning of Unmanned Aerial Systems for Visual Inspection of Power Transmission Lines and Towers. *IETE J. Res.* 2023, 70, 3259–3279.
8. Zhao, W.; Chu, H.; Zhang, M.; Sun, T.; Guo, L. Flocking Control of Fixed-Wing UAVs with Cooperative Obstacle Avoidance Capability. *IEEE Access* 2019, 7, 17798–17808.
9. Yu, Y.; Tingting, W.; Long, C.; Weiwei, Z. Stereo vision based obstacle avoidance strategy for quadcopter UAV. In *IEEE Industrial Electronics (IE) Chapter, Proceedings of the 2018 Chinese Control And Decision Conference (CCDC)*, Shenyang, China, 9–11 June 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 490–494.

10. Meng, Q.; Qu, Q.; Chen, K.; Yi, T. Multi-UAV Path Planning Based on Cooperative Co-Evolutionary Algorithms with Adaptive Decision Variable Selection. *Drones* 2024, 8, 435.
11. Debnath, D.; Hawary, A.F.; Ramdan, M.I.; Alvarez, F.V.; Gonzalez, F. QuickNav: An Effective Collision Avoidance and Path-Planning Algorithm for UAS. *Drones* 2023, 7, 678.
12. Vashisth, A.; Batth, R.S.; Ward, R. Existing Path Planning Techniques in Unmanned Aerial Vehicles (UAVs): A Systematic Review. In *Proceedings of the 2021 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE)*, Dubai, United Arab Emirates, 17–18 March 2021; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2021; pp. 366–372.
13. Elmeseiry, N.; Alshaer, N.; Ismail, T. A detailed survey and future directions of unmanned aerial vehicles (UAVs) with potential applications. *Aerospace* 2021, 8, 363.
14. Saadi, A.A.; Soukane, A.; Meraihi, Y.; Gabis, A.B.; Mirjalili, S.; Ramdane-Cherif, A. UAV Path Planning Using Optimization Approaches: A Survey. *Arch. Comput. Methods Eng.* 2022, 29, 4233–4284.
15. Puente-Castro, A.; Rivero, D.; Pazos, A.; Fernandez-Blanco, E. A review of artificial intelligence applied to path planning in UAV swarms. *Neural Comput. Appl.* 2022, 34, 153–170.
16. Aggarwal, S.; Kumar, N. Path planning techniques for unmanned aerial vehicles: A review, solutions, and challenges. *Comput. Commun.* 2020, 149, 270–299.
17. Zhao, Y.J.; Zheng, Z.; Liu, Y. Survey on computational-intelligence-based UAV path planning. *Knowl.-Based Syst.* 2018, 158, 54–64.
18. Dechter, R.; Pearl, J. Generalized Best-First Search Strategies and the Optimality of A*. *J. ACM* 1985, 32, 505–536.
19. Gobb, H.G.; Grefenstette, J.J. Genetic Algorithms for Tracking Changing Environments. In *Proceedings of the 5th International Conference on Genetic*

Algorithms, Urbana, IL, USA, 1 June 1993; Morgan Kaufmann Publishers Inc.: Burlington, MA, USA, 1993; pp. 523–530.

20. Eberhart, R.; Kennedy, J. A New Optimizer Using Particle Swarm Theory. In Proceedings of the Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, 4–6 October 1995.

21. Storn, R.; Price, K. Differential Evolution—A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *J. Glob. Optim.* 1997, 11, 341–359.

22. Colomi, A.; Dorigo, M.; Maniezzo, V. Distributed Optimization by Ant Colonies. In Proceedings of the First European Conference on Artificial Life, Paris, France, 11–13 December 1991; pp. 134–142.

23. Hart, P.E.; Nilsson, N.J.; Raphael, B. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Trans. Syst. Sci. Cybern.* 1968, 4, 100–107.

24. Mandloi, D.; Arya, R.; Verma, A.K. Unmanned aerial vehicle path planning based on A* algorithm and its variants in 3d environment. *Int. J. Syst. Assur. Eng. Manag.* 2021, 12, 990–1000.

25. Bai, X.; Jiang, H.; Cui, J.; Lu, K.; Chen, P.; Zhang, M. UAV Path Planning Based on Improved A* and DWA Algorithms. *Int. J. Aerosp. Eng.* 2021, 2021, 4511252.

26. Chen, J.; Li, M.; Yuan, Z.; Gu, Q. An Improved A* Algorithm for UAV Path Planning Problems. In Proceedings of the 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Chongqing, China, 12–14 June 2020; IEEE Press: Piscataway, NJ, USA, 2020; pp. 958–962.

27. Sun, S.; Wang, H.; Xu, Y.; Wang, T.; Liu, R.; Chen, W. A Fusion Approach for UAV Onboard Flight Trajectory Management and Decision Making Based on the Combination of Enhanced A* Algorithm and Quadratic Programming. *Drones* 2024, 8, 254.

28. Qi, D.; Zhang, Z.; Zhang, Q. Path Planning of Multirotor UAV Based on the Improved Ant Colony Algorithm. *J. Robot.* 2022, 2022, 2168964.
29. Li, G.; Li, Y. UAV path planning based on improved ant colony algorithm. In *Proceedings of the Second International Conference on Algorithms, Microchips, and Network Applications (AMNA 2023)*, Zhengzhou, China, 13–15 January 2023; pp. 59–63.
30. Huan, L.; Ning, Z.; Qiang, L. UAV Path Planning Based on an Improved Ant Colony Algorithm. In *Proceedings of the 2021 4th International Conference on Intelligent Autonomous Systems (ICoIAS)*, Wuhan, China, 14–16 May 2021; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2021; pp. 357–360.
31. Song, H.; Jia, M.; Lian, Y.; Fan, Y.; Liang, K. UAV Path Planning Based on an Improved Ant Colony Algorithm. *J. Electron. Res. Appl.* 2022, 6, 10–25.
32. Li, F.; Xiao, J.; Huang, W.; Cai, S. Research on the Intelligent Obstacle Avoidance and Path Planning Strategy of UAV based on Multi-Sensor Fusion. In *Proceedings of the 2022 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA)*, Dalian, China, 20–21 August 2022; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2022; pp. 628–632.
33. Wang, T.; Yang, L.; Chang, Y.; Huang, Z.; Jiang, H.; Zheng, Y. A Review of Dynamic Obstacle Avoidance for Unmanned Aerial Vehicles (UAVs). In *Proceedings of the 2024 7th International Symposium on Autonomous Systems (ISAS)*, Chongqing, China, 7–9 May 2024; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2024.
34. Sasongko, R.A.; Rawikara, S.S.; Tampubolon, H.J. UAV Obstacle Avoidance Algorithm Based on Ellipsoid Geometry. *J. Intell. Robot. Syst.* 2017, 88, 567–581.
35. Sandino, J.; Caccetta, P.A.; Sanderson, C.; Maire, F.; Gonzalez, F. Reducing Object Detection Uncertainty from RGB and Thermal Data for UAV Outdoor

Surveillance. In Proceedings of the IEEE Computer Society, Proceedings of the 2022 IEEE Aerospace Conference (AERO), Big Sky, MT, USA, 5–12 March 2022; IEEE Computer Society: Piscataway, NJ, USA, 2022.

36. Al-Kaff, A.; García, F.; Martín, D.; de la Escalera, A.; Armingol, J.M. Obstacle detection and avoidance system based on monocular camera and size expansion algorithm for UAVs. *Sensors* 2017, 17, 1061.

37. Aguilar, W.G.; Casalgilla, V.P.; Pólit, J.L. Obstacle avoidance based-visual navigation for micro aerial vehicles. *Electronics* 2017, 6, 10.

38. Lee, J.; Lee, K.; Park, S.; Im, S.; Park, J. Obstacle avoidance for small UAVs using monocular vision. *Aircr. Eng. Aerosp. Technol.* 2011, 83, 397–406.

39. Wang, H. UAV Obstacle Avoidance Technology. *Sci. Technol. Eng. Chem. Environ. Prot.* 2024, 1, 7.

40. Ardakani, S.P.; Cheshmehzangi, A. Reinforcement Learning-Enabled UAV Itinerary Planning for Remote Sensing Applications in Smart Farming. *Telecom* 2021, 2, 255–270.

41. Zou, K.; Wang, H.; Zhang, F.; Zhang, C.; Kai, D. Precision route planning method based on UAV remote sensing and genetic algorithm for weeding machine. *Appl. Intell.* 2023, 53, 11203–11213.

42. Mesquita, R.; Gaspar, P.D. A Novel Path Planning Optimization Algorithm Based on Particle Swarm Optimization for UAVs for Bird Monitoring and Repelling. *Processes* 2022, 10, 62.

43. Yao, C.; Zhang, Y.; Liu, H. Application of convolutional neural network in classification of high resolution agricultural remote sensing images. *ISPRS Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* 2017, XLII-2/W7, 989–992.

44. Vo, A.V.; Laefer, D.F.; Byrne, J. Optimizing urban LiDAR flight path planning using a genetic algorithm and a dual parallel computing framework. *Remote Sens.* 2021, 13, 4437.

45. Gupta, G.P.; Chawra, V.K.; Dewangan, S. Optimal path planning for UAV using NSGA-II based metaheuristic for sensor data gathering application in Wireless Sensor Networks. In Proceedings of the 2019 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS), Goa, India, 16–19 December 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–5.
46. Wang, M.; Zhu, C.; Wang, F.; Li, T.; Zhang, X. Multi-factor of path planning based on an ant colony optimization algorithm. *Ann. GIS* 2020, 26, 101–112.
47. Yang, Q.; Yoo, S.-J.; Yang, Q.; Yoo, S.-J. Optimal UAV Path Planning: Sensing Data Acquisition Over IoT Sensor Networks Using Multi-Objective Bio-Inspired Algorithms. *IEEE Access* 2018, 6, 13671–13684.
48. Sun, Z.; Wu, J.; Yang, J.; Huang, Y.; Li, C.; Li, D. Path Planning for GEO-UAV Bistatic SAR Using Constrained Adaptive Multiobjective Differential Evolution. *IEEE Trans. Geosci. Remote Sens.* 2016, 54, 6444–6457.
49. Ho, D.-T.; Grotli, E.I.; Sujit, P.B.; Johansen, T.A.; De Sousa, J.B. Performance evaluation of cooperative relay and Particle Swarm Optimization path planning for UAV and wireless sensor network. In Proceedings of the 2013 IEEE Globecom Workshops (GC Wkshps), Atlanta, GA, USA, 9–13 December 2013; IEEE: Piscataway, NJ, USA, 2013; pp. 1403–1408.
50. Cao, Z.; Jia, H.; Wang, Z.; Foh, C.H.; Tian, F. A differential evolution with autonomous strategy selection and its application in remote sensing image denoising. *Expert Syst. Appl.* 2024, 238, 122108.
51. Kestur, R.; Angural, A.; Bashir, B.; Omkar, S.N.; Anand, G.; Meenavathi, M.B. Tree Crown Detection, Delineation and Counting in UAV Remote Sensed Images: A Neural Network Based Spectral–Spatial Method. *J. Indian Soc. Remote Sens.* 2018, 46, 991–1004.
52. Aposporis, P. Object Detection Methods for Improving UAV Autonomy and Remote Sensing Applications. In Proceedings of the 2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), Hague,

The Netherlands, 7–10 December 2020; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2020; pp. 845–853.

53. Theile, M.; Bayerlein, H.; Nai, R.; Gesbert, D.; Caccamo, M. UAV coverage path planning under varying power constraints using deep reinforcement learning. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vegas, NV, USA, 24 October 2020–24 January 2021; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2021; pp. 1444–1449.

54. Cui, Z.; Wang, Y. UAV Path Planning Based on Multi-Layer Reinforcement Learning Technique. IEEE Access 2021, 9, 59486–59497.

55. Xie, R.; Meng, Z.; Wang, L.; Li, H.; Wang, K.; Wu, Z. Unmanned Aerial Vehicle Path Planning Algorithm Based on Deep Reinforcement Learning in Large-Scale and Dynamic Environments. IEEE Access 2021, 9, 24884–24900.

56. Gonzalez, V.; Monje, C.A.; Garrido, S.; Moreno, L.; Balaguer, C. Coverage Mission for UAVs Using Differential Evolution and Fast Marching Square Methods. IEEE Aerosp. Electron. Syst. Mag. 2020, 35, 18–29.

57. González, V.; Monje, C.; Moreno, L.; Balaguer, C. UAVs mission planning with flight level constraint using Fast Marching Square Method. Robot. Auton. Syst. 2017, 94, 162–171.

58. Lin, C.E.; Shao, P.C. Failure analysis for an unmanned aerial vehicle using safe path planning. J. Aerosp. Inf. Syst. 2020, 17, 358–369.