Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерної інженерії та інформаційних систем

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

Галузь знань 12 – Інформаційні технології

Спеціальність 123 –Комп'ютерна інженерія

на тему «Система планування та моніторингу руху групи БПЛА на основі аналізу параметрів їх компонентів»

КвРКІП. 301148.23.01.49 ПЗ

Виконав: студент 2 курсу, група КІ2м-23-1 _____ Максим ВЕЛИЧКО
Підпис Ім'я, прізвище

Керівник кандидат фіз.-мат. наук, доцент _____ Тетяна КИСІЛЬ
Науковий ступінь, вчене звання Підпис Ім'я, прізвище

До захисту допускаю:
Зав. кафедри КІПС, доктор філософії, доцент
Ольга ПАВЛОВА
22 05 2025 р.

Хмельницький, 2025

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет Інформаційних технологій

Кафедра Комп'ютерної інженерії та інформаційних систем

Освітній рівень магістр

Галузь знань 12 Інформаційні технології

Спеціальність 123 Комп'ютерна інженерія

Освітня програма освітньо-наукова програма «Комп'ютерна інженерія та програмування»

ЗАТВЕРДЖУЮ

Зав. кафедри Ольга ПАВЛОВА

"01" 09 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ МАГІСТРА

Максиму ВЕЛИЧКО

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Система планування та моніторингу руху групи БПЛА на основі аналізу параметрів їх компонентів

Керівник проекту (роботи) Тетяна КИСІЛЬ, к.ф.-м.н., доцент

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 08.01.2025 №8

2. Строк подання студентом проекту (роботи) на кафедру 01.05.2025 р.

3. Вихідні дані до проекту (роботи) Завдання на дипломне проектування

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

Огляд предметної області для системи планування та моніторингу руху групи БПЛА

Аналіз алгоритмів для планування та моніторингу маршрутів руху групи БПЛА

Моделі та методи планування траєкторій руху групи бпла в умовах неповної інформації

Моделювання системи планування руху для групи БПЛА

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень)

6. Консультанти розділів кваліфікаційної роботи магістра

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Сергій ЛИСЕНКО, професор кафедри КІІС		
Антиплагіат	Андрій НІЧЕПОРУК, доцент кафедри КІІС		

7. Дата видачі завдання « 01 » 09 2024р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) кваліфікаційної роботи магістра	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напряму дослідження та узгодження тематики КвРМ з керівником	01.09.2024	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.10.2024	виконано
3	Робота над розділом 1 – аналіз відомих моделей, методів за темою; постановка задачі	01.11.2024	виконано
4	Робота над розділом 2 – розробка моделей для вирішення поставленої задачі	01.12.2024	виконано
5	Робота над науковою статтею	01.02.2025	виконано
6	Робота над розділом 3 – розробка методів для вирішення поставленої задачі	15.02.2025	виконано
7	Робота над розділом 4 – проектування та розробка ПЗ для вирішення поставленої задачі, експериментальна частина	01.04.2025	виконано
8	Оформлення пояснювальної записки згідно вимог	18.04.2025	виконано
9	Попередній захист ДРМ	29.04.2025	виконано
10	Захист ДРМ на засіданні ЕК	До 23.05.2025	

Студент _____ Максим ВЕЛИЧКО
Підпис Ім'я, прізвище

Керівник роботи _____ Тетяна КИСІЛЬ
Підпис Ім'я, прізвище

РЕФЕРАТ

Тема кваліфікаційної роботи магістра: Система планування та моніторингу руху групи БПЛА на основі аналізу параметрів їх компонентів

Автор роботи: Величко Максим Олександрович

Керівник роботи: Кисіль Тетяна Миколаївна

Пояснювальна записка: 84 с., 38 рис., 3 табл., 2 дод., 99 джерел.

СИСТЕМА, БПЛА, ГРУПА БПЛА, ПЛАНУВАННЯ ТРАЄКТОРІЇ, МОНІТОРИНГ РУХУ, МАШИННЕ НАВЧАННЯ З ПІДКРІПЛЕННЯМ, БАГАТОАГЕНТНІ СИСТЕМИ, АЛГОРИТМИ МАРШРУТИЗАЦІЇ, ПРОКСИМАЛЬНА ОПТИМІЗАЦІЯ ПОЛІТИКИ, ЦЕНТРАЛІЗОВАНЕ НАВЧАННЯ ТА ДЕЦЕНТРАЛІЗОВАНЕ ВИКОНАННЯ.

Об’єктом дослідження є процес планування та моніторингу руху групи безпілотних літальних апаратів (БПЛА).

Предметом дослідження є методи та алгоритми планування траєкторій і моніторингу руху групи БПЛА на основі аналізу параметрів їх компонентів.

Метою кваліфікаційної роботи магістра є покращення процесів планування та моніторингу руху групи БПЛА шляхом розроблення відповідної системи, яка ґрунтується на аналізі параметрів компонентів БПЛА для забезпечення оптимального виконання польотних завдань та підвищення ефективності роботи групи.

Для розв’язання поставлених задач використовувалися методи теоретичного моделювання багатоагентних систем та алгоритмічного аналізу Основою обраного підходу стало застосування методів машинного навчання з підкріпленням, зокрема алгоритму проксимальної оптимізації політики (PPO), заснований на акторно-критичній структурі. Для забезпечення ефективного навчання в умовах багатьох агентів використовувалася схема централізованого навчання з децентралізованим виконанням (CTDE). З метою врахування часових залежностей і вирішення задачі прийняття рішень в умовах неповної інформації використовувався шар рекурентної

нейронної мережі (RNN), яка забезпечує обробку історичних даних агентів при формуванні політики управління.

Наукова новизна отриманих результатів:

− удосконалено метод проксимальної оптимізації політики (PPO) для планування та моніторингу руху групи БПЛА в умовах неповної інформації;

− удосконалено структуру нейронної мережі актора та критика алгоритму проксимальної оптимізації політики (PPO) за допомогою шару рекурентної нейронної мережі (RNN), яка забезпечує обробку історичних даних агентів при формуванні політики управління.

На основі проведених досліджень розроблений багатоагентний алгоритм проксимальної оптимізації політики (PPO) в багатоагентному середовищі для планування та моніторингу руху групи БПЛА. Для того, щоб вирішити проблему прийняття рішень з неповною інформацією, було обрано архітектуру CTDE і додано шар RNN для використання історичної інформації. Крім того, розроблено спільну функцію винагороди, щоб допомогти групам БПЛА навчитися оптимальній політиці. Система протестована в умовах, що моделюють реальні сценарії роботи групи БПЛА, що дозволило оцінити ефективність запропонованих підходів до управління та координації руху в динамічному середовищі.

Практична значимість отриманих результатів полягає у підвищенні ефективності роботи групи БПЛА завдяки впровадженню покращеного методу планування та моніторингу руху. Створена система здатна автоматично планувати маршрути, координувати роботу групи БПЛА, здійснювати моніторинг стану компонентів та адаптуватися до змін у реальному часі. Система може бути інтегрована в існуючі платформи для управління БПЛА, забезпечуючи автоматизований контроль та оптимізацію траєкторій. У результаті це дозволить підвищити точність виконання завдань та поліпшити ефективність виконання складних місій, таких як розвідка або доставка.

У першому розділі проведено огляд можливостей застосування БПЛА в різних сферах, проаналізовано сучасні технології енергозабезпечення, методи

планування та моніторингу групових польотів, а також визначено технічні виклики й сформульовано постановку задачі.

У другому розділі здійснено аналіз методів машинного навчання, таких як контрольоване, неконтрольоване та навчання з підкріпленням. Проведено порівняння відповідних алгоритмів у контексті планування маршрутів БПЛА.

У третьому розділі реалізовано підхід на основі алгоритму проксимальної оптимізації політики (PPO) у поєднанні з архітектурою централізованого навчання та децентралізованого виконання (CTDE), що дозволяє дронам ефективно навчатися в умовах часткової спостережуваності середовища. Додано рекурентний шар (RNN, зокрема LSTM) в структуру мереж, що дозволило зменшити вплив неповної інформації, покращити узагальнення та стабільність навчання.

У четвертому розділі було проведено моделювання та емпіричну оцінку ефективності системи планування руху для групи безпілотних літальних апаратів (БПЛА) з використанням навчання з підкріпленням на основі алгоритму проксимальної оптимізації політики (PPO) в архітектурі централізованого навчання з децентралізованим виконанням (CTDE).

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ 5

ВСТУП 6

1 ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ ДЛЯ СИСТЕМИ ПЛАНУВАННЯ ТА МОНІТОРИНГУ РУХУ ГРУПИ БПЛА НА ОСНОВІ АНАЛІЗУ ПАРАМЕТРІВ ЇХ КОМПОНЕНТІВ 9

1.1 Аналіз характеристик, компонентів та сфер застосування БПЛА для планування та моніторингу їх руху 9

1.2 Сучасні технології енергозабезпечення та підзарядки БПЛА 18

1.3 Огляд підходів до планування та моніторингу руху групи БПЛА 23

1.4 Основні виклики та технічні обмеження у плануванні й моніторингу групових польотів БПЛА 24

1.5 Сучасні підходи до управління та моніторингу руху групи БПЛА 26

1.6 Постановка задачі 28

1.7 Висновки до першого розділу 29

2 АНАЛІЗ АЛГОРИТМІВ ДЛЯ ПЛАНУВАННЯ ТА МОНІТОРИНГУ МАРШРУТІВ РУХУ ГРУПИ БПЛА 31

2.1 Машинне навчання 31

2.1.1 Контрольоване машинне навчання 32

2.1.2 Неконтрольоване машинне навчання 42

2.1.3 Машинне навчання з підкріпленням 46

2.2 Порівняння приведених алгоритмів 56

2.3 Висновки до другого розділу 58

3 МОДЕЛІ ТА МЕТОДИ ПЛАНУВАННЯ ТРАЄКТОРІЙ РУХУ ГРУПИ БПЛА В УМОВАХ НЕПОВНОЇ ІНФОРМАЦІЇ 59

3.1 Багатоагентне планування маршрутів: постановка задачі та виклики 59
3.2 Використання навчання з підкріпленням у плануванні маршрутів БПЛА 59
3.3 Алгоритм актор-критик..... 63
3.4 Централізоване навчання та децентралізована архітектура виконання 64
3.5 Проксимальна оптимізація політики за допомогою CTDE..... 65
3.6 Додавання шару RNN для неповної інформації..... 68
3.7 Багатоцільова оптимізація 72
3.8 Висновки до третього розділу 76
4 МОДЕЛЮВАННЯ СИСТЕМИ ПЛАНУВАННЯ РУХУ ДЛЯ ГРУПИ БПЛА 78
4.1 Архітектура мережі..... 78
4.2 Проведення моделювання та тестування..... 80
4.3 Висновки до четвертого розділу..... 86
ВИСНОВКИ 88
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ 90
ДОДАТОК А Презентаційні матеріали **..... 100**
ДОДАТОК Б Наукова праця здобувача**..... 107**

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

БПЛА – безпілотний літальний апарат

GPS – Global Positioning System

MAPF – багатоагентне планування маршруту

DRL – глибоке навчання з підкріпленням

PPO – проксимальна оптимізація політики

Dec-POMDP – децентралізовани частковий спостережуваний марковський процес прийняття рішень

CTDE – архітектура централізованого навчання з децентралізованим виконанням

RNN – рекурентні нейронні мережі

DRQN – глибоке рекурентне Q-навчання

LSTM – архітектура довготривалої короткочасної пам'яті

MGV – мобільні наземні транспортні засоби

ВСТУП

У сучасному світі, де автоматизація процесів стає невід’ємною частиною, безпілотні літальні апарати набувають все більшого застосування в різних галузях, таких як сільське господарство, екологія, розважальна індустрія, в роботі правоохоронних органів та армії. В Україні БПЛА набули широкого використання у військовому напрямку і швидко розвиваються. А як показала історія, багато технологій, розроблених для військових цілей, згодом знаходять своє застосування в цивільних сферах. Зокрема, в Україні існують великі перспективи для використання БПЛА в сільському господарстві, адже аграрний бізнес займає важливу роль в економіці країни.

У зв'язку з широким застосуванням груп БПЛА, зростає необхідність у розробці ефективних систем для їх управління, планування траєкторій та моніторингу їхнього руху. Задача управління групами БПЛА є складною, оскільки кожен апарат має свої індивідуальні параметри, такі як висота польоту, швидкість, залишковий заряд батареї, потужність двигуна та інші характеристики, що можуть змінюватися під впливом зовнішніх факторів.

Саме тому виникає необхідність у розробці системи, яка буде аналізувати ці параметри в реальному часі та оптимізувати рух БПЛА для забезпечення їх максимальної ефективності та безпеки.

Об’єктом дослідження є процес планування та моніторингу руху групи безпілотних літальних апаратів (БПЛА).

Предметом дослідження є методи та алгоритми планування траєкторій і моніторингу руху групи БПЛА на основі аналізу параметрів їх компонентів.

Метою кваліфікаційної роботи магістра є покращення процесів планування та моніторингу руху групи БПЛА шляхом розроблення відповідної системи, яка ґрунтується на аналізі параметрів компонентів БПЛА для забезпечення оптимального виконання польотних завдань та підвищення ефективності роботи групи.

Для розв’язання поставлених задач використовувалися методи теоретичного моделювання багатоагентних систем та алгоритмічного аналізу Основою обраного підходу стало застосування методів машинного навчання з підкріпленням, зокрема алгоритму проксимальної оптимізації політики (PPO), заснований на акторно-критичній структурі. Для забезпечення ефективного навчання в умовах багатьох агентів використовувалася схема централізованого навчання з децентралізованим виконанням (CTDE). З метою врахування часових залежностей і вирішення задачі прийняття рішень в умовах неповної інформації використовувався шар рекурентної нейронної мережі (RNN), яка забезпечує обробку історичних даних агентів при формуванні політики управління.

Наукова новизна отриманих результатів:

− удосконалено метод проксимальної оптимізації політики (PPO) для планування та моніторингу руху групи БПЛА в умовах неповної інформації;

− удосконалено структуру нейронної мережі актора та критика алгоритму проксимальної оптимізації політики (PPO) за допомогою шару рекурентної нейронної мережі (RNN), яка забезпечує обробку історичних даних агентів при формуванні політики управління.

На основі проведених досліджень розроблений багатоагентний алгоритм проксимальної оптимізації політики (PPO) в багатоагентному середовищі для планування та моніторингу руху групи БПЛА. Для того, щоб вирішити проблему прийняття рішень з неповною інформацією, було обрано архітектуру CTDE і додано шар RNN для використання історичної інформації. Крім того, розроблено спільну функцію винагороди, щоб допомогти групам БПЛА навчитися оптимальній політиці. Система протестована в умовах, що моделюють реальні сценарії роботи групи БПЛА, що дозволило оцінити ефективність запропонованих підходів до управління та координації руху в динамічному середовищі.

Практична значимість отриманих результатів полягає у підвищенні ефективності роботи групи БПЛА завдяки впровадженню покращеного методу планування та моніторингу руху. Створена система здатна автоматично планувати маршрути, координувати роботу групи БПЛА, здійснювати моніторинг стану

компонентів та адаптуватися до змін у реальному часі. Система може бути інтегрована в існуючі платформи для управління БПЛА, забезпечуючи автоматизований контроль та оптимізацію траєкторій. У результаті це дозволить підвищити точність виконання завдань та поліпшити ефективність виконання складних місій, таких як розвідка або доставка.

Отже застосування такої системи сприятиме підвищенню ефективності та безпеки безпілотних систем. У перспективі така система відкриє нові можливості для розвитку технологій управління групами БПЛА, що матиме значний вплив на майбутній розвиток безпілотних технологій як в Україні, так і в світі.

За темою кваліфікаційної роботи опубліковано одну статтю [99] у фаховому науковому журналі: VELYCHKO M., KYSIL T. M. REINFORCEMENT LEARNING METHOD FOR AUTONOMOUS FLIGHT PATH PLANNING OF MULTIPLE UAVS. *Computer Systems and Information Technologies*. 2025. №. 2.

1 ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ ДЛЯ СИСТЕМИ ПЛАНУВАННЯ ТА МОНІТОРИНГУ РУХУ ГРУПИ БПЛА НА ОСНОВІ АНАЛІЗУ ПАРАМЕТРІВ ЇХ КОМПОНЕНТІВ

1.1 Аналіз характеристик, компонентів та сфер застосування БПЛА для планування та моніторингу їх руху

Безпілотні літальні апарати (БПЛА) набувають все більшого поширення у різних сферах діяльності завдяки їхній мобільності, автономності та можливості виконувати складні завдання в умовах, недоступних для людини. Сучасні технології дозволяють застосовувати групи БПЛА для виконання колективних місій, таких як аерофотозйомка, моніторинг територій, розвідка, логістика та пошуково-рятувальні операції [1].

Дрони стали важливим інструментом у цивільній та комерційній сферах, зокрема в умовах міського середовища, у так званій концепції «розумного міста» [2], забезпечуючи надання широкого спектра послуг, зокрема у сфері спостереження та безпеки. Виконуючи функцію мобільних камер, БПЛА ефективно здійснюють моніторинг за об’єктами інфраструктури та скупченням людей, сприяючи органам влади в управлінні масовими заходами на вулицях та в громадських місцях.

Завдяки високій мобільності безпілотники можуть здійснювати відеозйомку з різних ракурсів, надаючи цінну інформацію щодо динаміки переміщення натовпу та перебігу подій [3]. Це, у свою чергу, сприяє оперативному виявленню та аналізу надзвичайних ситуацій, а також дозволяє здійснювати ефективне відстеження осіб, які викликають підозру.

Окрім цього, БПЛА здатні здійснювати збір інформації про стан дорожнього руху з метою аналізу транспортних потоків та виявлення заторів у міській мережі. Також можливий збір даних щодо погодних умов та рівня забруднення навколишнього середовища. Отримана інформація може бути використана для оптимізації міського планування та раціонального розподілу ресурсів.

Ще одним важливим напрямом застосування безпілотних літальних апаратів у концепції «розумного міста» є доставка вантажів [4] (рисунок 1.1). Значна кількість наукових досліджень присвячена розробленню ефективних систем доставки із використанням БПЛА. Зокрема, наводиться перелік із 27 країн, які впровадили безпілотні технології для транспортування різноманітних товарів – від продуктів харчування до медичних засобів.

Цей інноваційний підхід до логістики забезпечує підвищену швидкість доставки, зменшення витрат і покращення доступності послуг у межах міської інфраструктури.

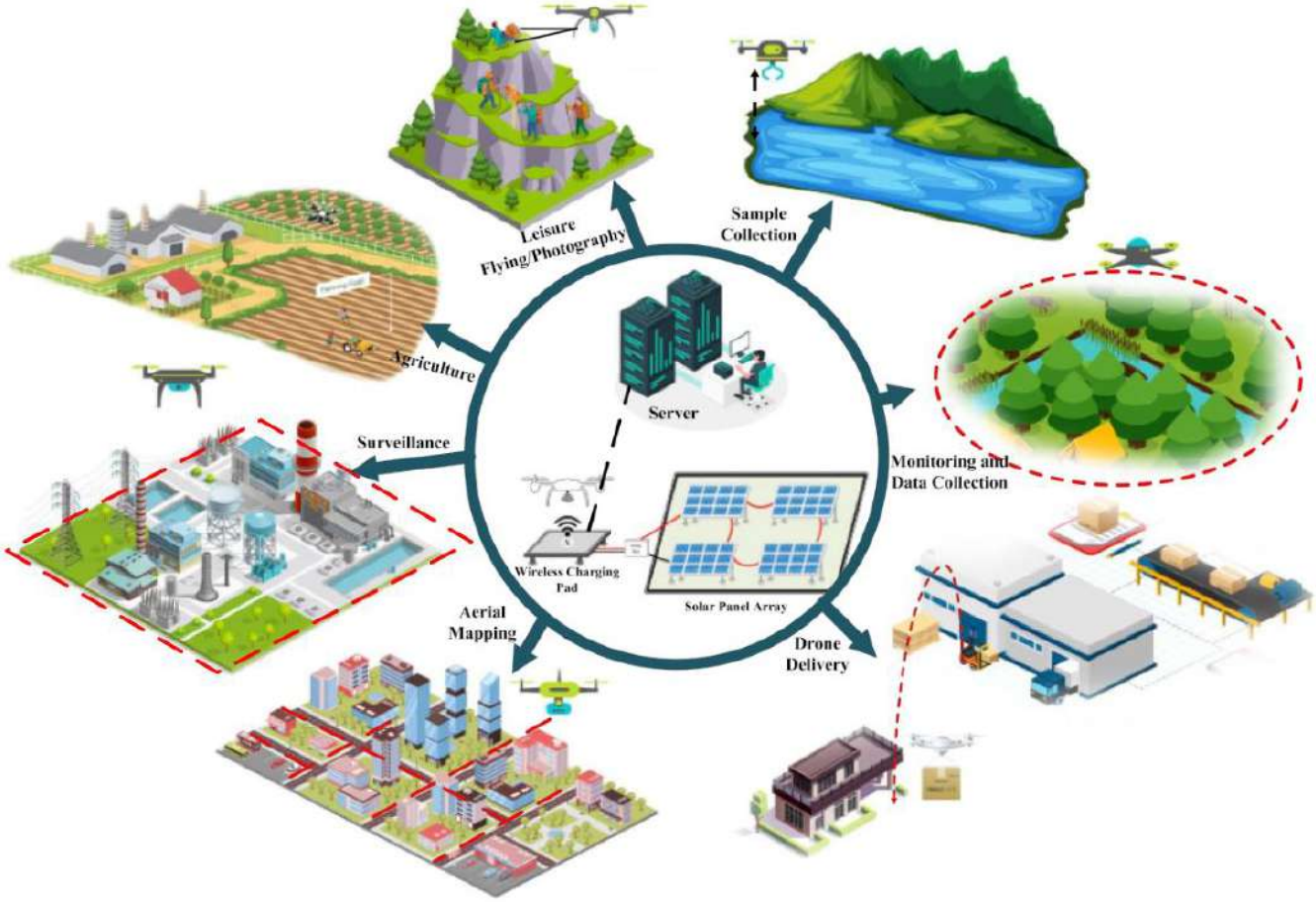


Рисунок 1.1 – Застосування комерційних дронів у різних секторах [4]

Крім того, БПЛА демонструють високу ефективність на будівельних та промислових об’єктах. Вони здатні транспортувати матеріали та інструменти, здійснювати нагляд за робочими процесами, виконувати дистанційний технічний

огляд інфраструктурних об’єктів, а також збирати важливі аналітичні дані щодо стану будівництва або промислового майданчика. Завдяки використанню дронів у цих середовищах управління об'єктами стає більш ефективним, безпечним, економічно вигідним і масштабованим [5].

У сфері «розумного» сільського господарства застосування БПЛА має численні переваги. Ці технологічні засоби здатні здійснювати збір важливої інформації щодо стану ґрунтів, росту та загального стану сільськогосподарських культур (рисунок 1.1), що дозволяє агровиробникам ухвалювати обґрунтовані управлінські рішення та своєчасно реагувати на потенційні проблеми.

Завдяки впровадженню безпілотних технологій фермери отримують можливість оптимізувати використання ресурсів, зменшити виробничі витрати та підвищити врожайність, що в результаті сприяє досягненню як економічних, так і екологічних ефектів. Наприклад, ще у 2007 році в Японії безпілотні вертольоти вже використовувалися для обприскування понад 10 % рисових плантацій [6].

Окрім цього, БПЛА є ефективним інструментом для відстеження тварин на «розумних» фермах та у великих лісових масивах. Їхнє використання значно полегшує трудомісткий процес пошуку та моніторингу диких тварин [7], забезпечуючи дослідників і природоохоронні організації точними й актуальними даними.

Завдяки застосуванню дронів дослідники можуть отримувати інформацію про поведінкові особливості тварин, маршрути міграції та динаміку популяцій [8]. Ці відомості є важливою складовою у реалізації заходів із збереження біорізноманіття та розроблення ефективних стратегій управління природними ресурсами.

Безпілотні апарати також довели свою ефективність у морських просторах: вони здатні забезпечувати критично важливу інформацію для спостереження за морською фауною, моніторингу наслідків забруднення та управління надзвичайними ситуаціями в океанічному середовищі [9].

Застосування безпілотних літальних апаратів у сільському господарстві та сфері моніторингу навколишнього природного середовища постійно розвивається.

Вдосконалення технологій БПЛА – зокрема, збільшення тривалості автономної роботи, впровадження автономної навігації [10] та застосування передових алгоритмів аналізу даних – здатне суттєво трансформувати практики агровиробництва та заходи із збереження довкілля.

Втім, для повноцінного використання потенціалу дронів у зазначених галузях необхідно розв'язати низку проблем, зокрема: регуляторні обмеження [11], питання конфіденційності даних [12], а також технічні бар'єри, пов'язані з надійністю [13] та масштабованістю [14] таких систем.

У надзвичайних ситуаціях, зокрема під час стихійних лих або в умовах загроз безпеці, дрони відіграють ключову роль у системах реагування. Їх використання дозволяє компенсувати дефіцит людських ресурсів та обмеження, спричинені пошкодженням інфраструктури. БПЛА забезпечують оперативну оцінку постраждалих територій [15], сприяють ефективному проведенню пошуково-рятувальних операцій та виявленню потенційних загроз [16].

Більшість вищезазначених сфер застосування дронів прямо або опосередковано пов'язані з управлінням ризиками та реагуванням на надзвичайні події. Наприклад, на промислових об'єктах використання БПЛА для моніторингу технічного стану інфраструктури [17] знижує ймовірність виникнення аварійних ситуацій.

Технології, які використовуються для спостереження за дикою природою у лісових масивах, можуть бути адаптовані для виявлення та моніторингу лісових пожеж [18]. Утім, успішна реалізація такого підходу потребує адаптації наявних систем до специфічних умов нових сценаріїв застосування.

Окрім функцій збору даних, безпілотні літальні апарати можуть виконувати роль систем доставки у надзвичайних ситуаціях, забезпечуючи транспортування життєво необхідних вантажів до віддалених або ізольованих населених пунктів [19]. Така можливість є особливо важливою у важкодоступних регіонах, де традиційні засоби транспортування можуть бути недоступними через пошкоджену інфраструктуру або складні умови пересування.

Застосування БПЛА для оперативної доставки ресурсів дозволяє рятувальним службам швидко надавати допомогу постраждалому населенню, зокрема забезпечуючи доставку медикаментів, продуктів харчування чи крові [20]. Яскравим прикладом ефективності такого підходу є система доставки донорської крові у Республіці Руанда [21], яка вже сприяла збереженню численних людських життів.

Однією з ключових проблем при використанні груп БПЛА є ефективне планування траєкторій руху та моніторинг їх стану в реальному часі [22]. Це завдання ускладнюється через різнорідність характеристик кожного апарата, змінність зовнішніх умов, а також необхідність координації між кількома літальними апаратами.

Важливим аспектом є врахування параметрів компонентів БПЛА, таких як рівень заряду батареї, потужність двигуна, швидкість, висота польоту, температура компонентів та інші характеристики [23]. Ці параметри безпосередньо впливають на працездатність кожного дрона та групи в цілому. Аналіз даних параметрів у реальному часі дозволяє оптимізувати траєкторії, рівномірно розподіляти навантаження між БПЛА та своєчасно приймати рішення про повернення апаратів на базу для підзарядки або заміни [24].

На сьогодні існують різні методи планування руху груп БПЛА [25], зокрема:

– теорія графів;

– методи розв’язання задачі комівояжера та задачі кількох комівояжерів у межах теорії транспортних систем;

– математичні оптимізаційні моделі, зокрема еволюційні алгоритми;

– алгоритми пошуку найкоротших шляхів (A*, Дейкстри);

– методи ройового інтелекту (Particle Swarm Optimization, Ant Colony Optimization);

– централізовані системи керування.

Проте більшість існуючих рішень не враховують динамічні зміни параметрів компонентів під час виконання місій, що може призводити до неефективного використання ресурсів або навіть виходу апаратів з ладу.

Через значну кількість БПЛА, що активно розробляються та використовуються в останні роки, постає проблема їх класифікації [26]. Оскільки ці апарати застосовуються в різних галузях, створити універсальну систему класифікації, яка б охоплювала всі типи БПЛА, досить складно. Внаслідок цього на сьогодні не існує єдиного стандарту для їх класифікації. Оборонні відомства користуються власними стандартами, тоді як цивільні розробники розробляють свої категорії, які постійно вдосконалюються.

Загалом БПЛА класифікують за двома основними критеріями: експлуатаційними характеристиками та їх призначенням.

До експлуатаційних характеристик відносяться: вага, дальність і тривалість польоту, корисне навантаження, вартість, швидкість, тип двигуна та його потужність.

Найпоширенішими аспектами призначення є цілевизначення та розвідка, вертикальний зліт і посадка, використання у вигляді радара або ретранслятора зв’язку, а також логістичні завдання, зокрема повітряна доставка [27, 28].

Класифікація БПЛА [29] може ґрунтуватися на конфігурації їх крил як нерухоме, гвинтокриле та гібридне. На відміну від БПЛА з нерухомим крилом, БПЛА з гвинтокрилом менш дорогі та забезпечують великі переваги для легкого зльоту, посадки та зависання; однак вони мають обмежену витривалість і швидкість порівняно з БПЛА з нерухомим крилом. Вони використовують обертові гвинти для створення тяги та контролю підйомної сили, подібно до вертольотів. Багатороторні БПЛА є найпопулярнішим типом гвинтокрилих БПЛА з двома або більше гвинтами. Вони широко використовуються в багатьох цивільних комерційних програмах для моніторингу та інспекції, оскільки вони мають чудові можливості виконувати швидкі повороти, маневри та підтримувати швидкість. Легкість планування їхнього шляху, що має менше обмежень порівняно з БПЛА з нерухомим крилом, а також здатність легко та швидко злітати й приземлятися спеціально на транспортні засоби, що рухаються, робить їх гідними для супроводу та підтримки мобільних наземних транспортних засобів (MGV) у негайних

розвідувальних місіях і місіях із прогнозуванням для забезпечення розширеного повітряного зондування та покриття, що перевищує можливості MGV.

Існує багато видів дронів різних форм і розмірів. Деякі приклади показано на рисунку 1.2. Ми можемо розділити їх на різні категорії залежно від розміру, форми крила, ваги, висоти польоту, дальності, автономності, силової установки та застосування [30].

(а) Багатороторні дрони (б) Дрони з нерухомим крилом (в) Різноманітні форми

Рисунок 1.2 – Приклади типів БПЛА [30]

Безпілотники можна класифікувати за типом крил, з двома основними категоріями: багатороторні дрони [31] та дрони з нерухомим крилом [32]. Багатороторні дрони, зображені на рисунку 1.2(а), часто зустрічаються, і їх миттєво впізнають як дрони. Вони популярні завдяки доступності, простоті використання, винятковій стійкості та маневреності. Ці безпілотники можуть легко злітати та приземлятися вертикально, підтримувати стабільне зависання та швидко переміщатися у вузьких просторах. Хоча час їх польоту та вантажопідйомність зазвичай обмежені, вони можуть сильно відрізнятися. Дальність їх польоту може варіюватися від 20 м до 10 км, що сильно впливає на їх ціну.

З іншого боку, є дрони з нерухомим крилом, зображені на рисунку 1.2(б). Конструкцією та функціями вони дуже нагадують літаки. Вони покладаються на рухову систему, щоб рухатися вперед, а крила створюють необхідну підйомну силу для тривалого польоту. Отже, ці дрони вимагають достатнього простору для зльоту

та посадки, але є більш енергоефективними, ніж їхні багатороторні аналоги. Вони можуть розвивати більш високу швидкість і мати більшу дальність польоту. Однак дрони з нерухомим крилом, як правило, дорожчі. Існують додаткові варіації, які ми бачимо не так часто, як-от одноторні БПЛА, що нагадують гелікоптери, і гібридні, які поєднують характеристики як багатороторних, так і безпілотних літальних апаратів з нерухомим крилом (рисунок 1.2(в)).

Типові БПЛА складаються з різних компонентів (рисунок 1.3), а саме з польотного контролера (FC), безщіткових двигунів постійного струму (BLDC), електронного регулятора швидкості (ESC), плати розподілу живлення (PDB), акумулятора, радіопередавача-приймача, камери з видом від першого лиця (FPV), модуля передачі-приймання відео та рами [33]. Вони забезпечують виконання польотних завдань, стабільність, комунікацію та обробку даних.

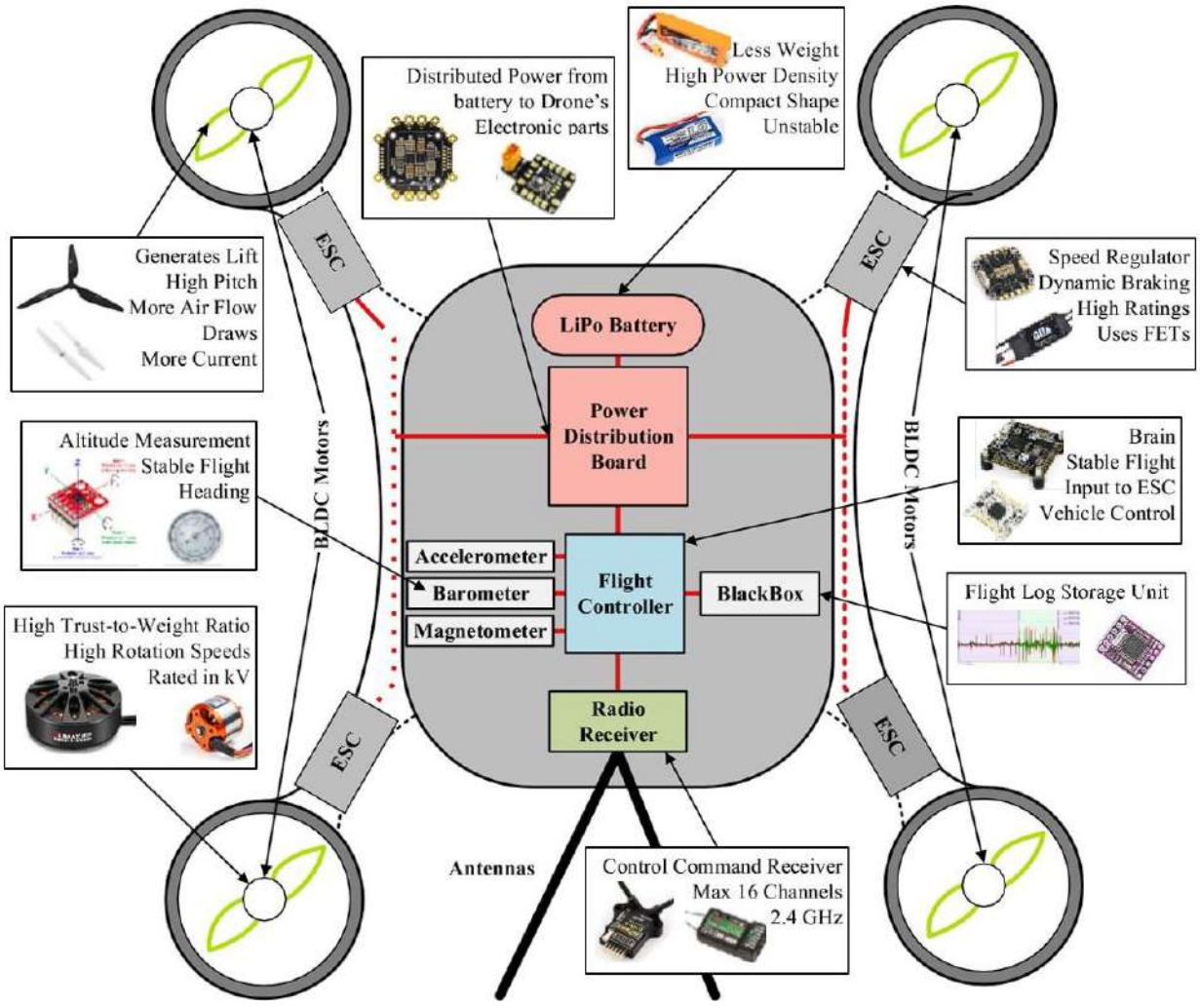


Рисунок 1.3 – Внутрішнє обладнання типового дрона [33]

Основою функціонування будь-якого БПЛА є бортовий комп'ютер, який обробляє команди та координує роботу всіх систем. Контролер польоту (FC) – є мозком системи і відповідає за стабільність, керування двигунами та зберігання бортового журналу. На рисунку 1.4 показано як він використовує пропорційно-інтегрально-диференціальний (ПІД) регулятор для зчитування сигналів, отриманих від радіоприймача, як зміну кутових швидкостей для керування напрямком польоту дрона. Пропорційний, інтегральний та диференціальний блоки мають відповідні масштабні коефіцієнти, які необхідно налаштувати для ефективного керування польотом. Потім модифікований сигнал надсилається до відповідних електронних регуляторів швидкості (ESC) для керування рухом дрона [34]. Радіоприймач зазвичай працює на частоті 2,45/5,8 ГГц залежно від навколишнього середовища та перешкод між наземною системою та дроном. Частота 5,8 ГГц має великий радіус дії, але обмежені можливості передачі даних, і навпаки.

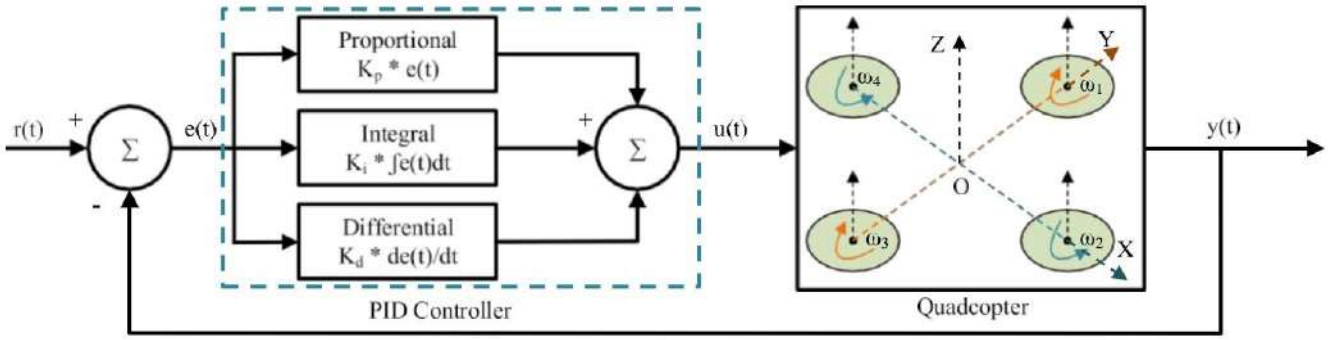


Рисунок 1.4 – Функціональна блок-схема контуру ПІД-регулятора [35]

Важливим елементом БПЛА є різноманітні датчики, що забезпечують орієнтацію та безпеку польоту. Акселерометри та гіроскопи допомагають підтримувати стабільність, барометричні датчики визначають висоту, а магнітометри дозволяють орієнтуватися по напрямку. Для обльоту перешкод можуть використовуватися оптичні або ультразвукові датчики, які дозволяють здійснювати безпечний рух у складних умовах.

Силова установка включає BLDC двигуни – це компактні потужні двигуни, які працюють на високих обертах та відповідають за підйомну силу та рух БПЛА,

а керування ними здійснюється за допомогою регуляторів обертів (ESC) шляхом подачі необхідної потужності від джерела до двигуна за допомогою керуючих сигналів від бортового контролера, що контролюють швидкість обертання пропелерів.

Джерелом енергії є акумуляторні батареї, такі як свинцево-кислотні (Lead Acid), нікель-метал-гідридні (NiMH) та літієві батареї, зокрема літій-іонні (Li-ion) та літій-полімерні (Li-Po), які забезпечують високу ємність та ефективність [35]. Ефективність роботи БПЛА напряму залежить від рівня заряду батареї та температури компонентів, що потребує постійного моніторингу.

Комунікаційні системи забезпечують зв'язок БПЛА із наземними станціями або іншими дронами в групі. Для цього використовуються радіомодулі, телеметрійні системи та GPS-антени, які передають дані про стан апарата та дозволяють коригувати маршрут у реальному часі. Сучасні БПЛА часто оснащені камерами для відеозйомки, тепловізорами або лідарами, що використовуються для моніторингу місцевості, пошукових операцій або картографії.

FPV камера використовується для отримання досвіду керування дроном з точки зору сидіння всередині дрона. Це корисно під час керування дроном поза межами прямої видимості. FPV камера працює за допомогою модуля передачі-приймача відеосигналу і живиться від акумулятора дрона.

Програмне забезпечення відіграє ключову роль у керуванні БПЛА [36]. Воно дозволяє планувати маршрути, оптимізувати польотні завдання та обробляти отримані дані. Для автоматизованого моніторингу використовуються системи комп'ютерного зору, які здатні розпізнавати об'єкти та оцінювати обстановку. Додатково в БПЛА інтегруються системи контролю енергоспоживання, які оптимізують витрати енергії та продовжують час автономного польоту.

1.2 Сучасні технології енергозабезпечення та підзарядки БПЛА

Найпоширенішим способом підзаряджання безпілотних літальних апаратів (БПЛА) залишається заміна акумуляторних батарей. Водночас науково-технічна

спільнота запропонувала низку інноваційних підходів до забезпечення енергоживлення дронів. Методи підзаряджання [37], як правило, класифікують за двома основними групами: із використанням електромагнітного поля (EMF) та без його застосування.

До першої групи належить метод Battery Dumping (рисунок 1.5(а)), що передбачає оснащення БПЛА кількома акумуляторними батареями. У процесі польоту розряджена батарея автоматично від'єднується або скидається, що сприяє зменшенню загальної маси апарата й подовженню часу його автономної роботи.

Іншим способом є інтеграція високоефективних фотоелектричних панелей у конструкцію дрона (рисунок 1.5(б)). У світлу пору доби ці панелі забезпечують безпосереднє живлення двигунів, тоді як накопичена енергія використовується в нічний час. Основним недоліком цього підходу є залежність від інтенсивності сонячного випромінювання, що істотно знижує ефективність роботи дрона за несприятливих погодних умов або в темний період доби.

Серед альтернативних методів варто виокремити технологію Gust Soaring (ширяння у повітряних потоках), за аналогією до польоту альбатроса (рисунок 1.5(в)). Цей метод полягає в оптимізації траєкторії руху БПЛА з фіксованим крилом, щоб апарат ефективно використовував висхідні повітряні потоки та зустрічний вітер для збереження енергії, практично не витрачаючи її на підтримання польоту.

Ще одним перспективним підходом є лазерне заряджання (рисунок 1.5(г)). Суть методу полягає в передачі енергії за допомогою інфрачервоного лазерного випромінювання, спрямованого на модифікований сонячний елемент, вмонтований у нижню частину корпусу дрона. У межах експерименту квадрокоптер, що використовував цей спосіб підзарядки, продемонстрував здатність до безперервного польоту протягом 12 годин.

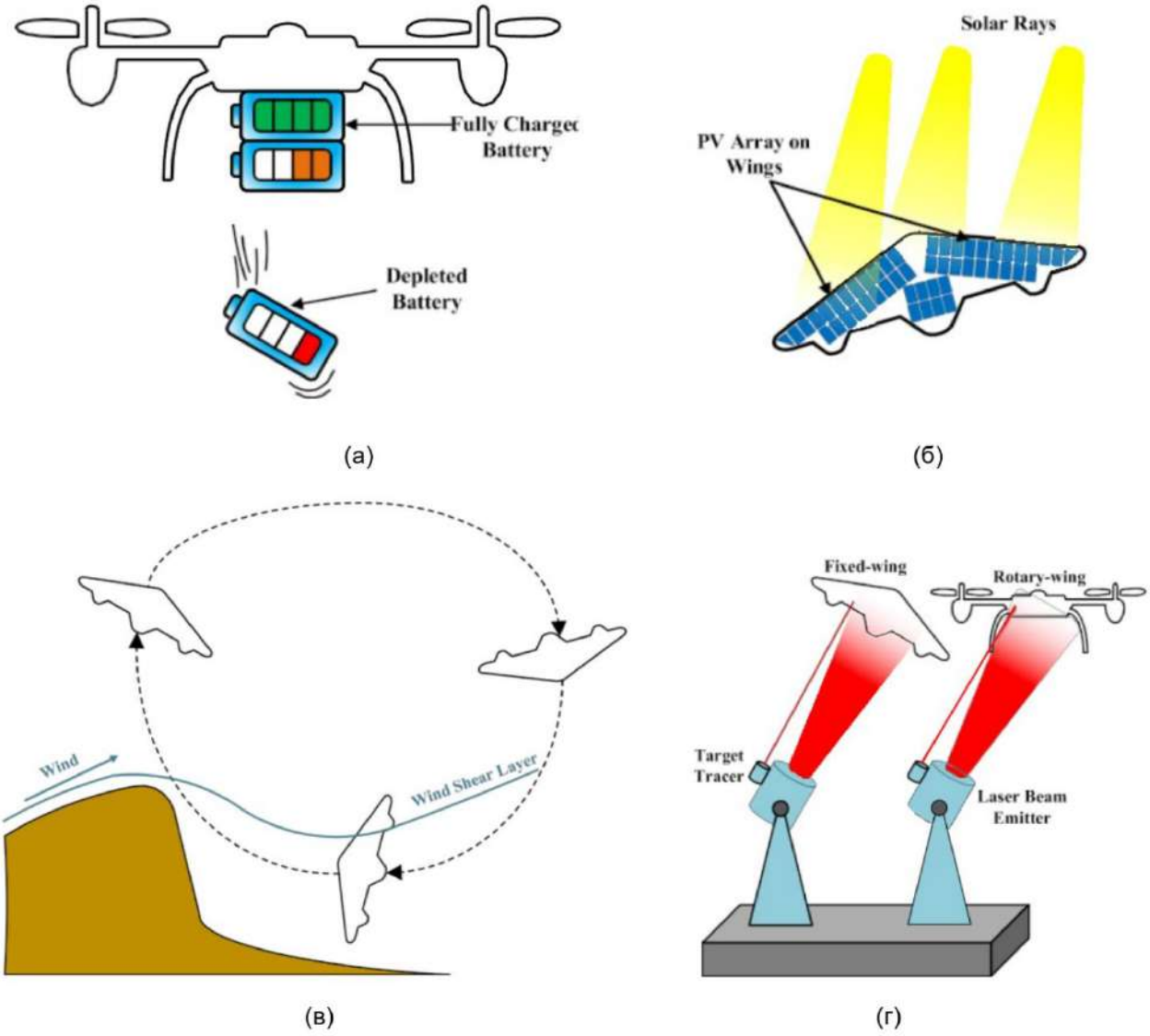


Рисунок 1.5 – Методи зарядки акумуляторів БПЛА: (а) скидання батареї (б) встановлення фотоелектричних панелей на крилах БПЛА (в) ширяння у повітряних потоках (г) лазерний промінь [35]

Попри інноваційність наведених методів, більшість із них мають низку обмежень. Зокрема, вони малопридатні для квадрокоптерів, а також не забезпечують необхідного рівня автономності під час виконання завдань у визначених зонах інтересу.

У цьому контексті бездротова передача енергії (Wireless Power Transfer, WPT) розглядається як оптимальне рішення для забезпечення безперервного енергоживлення дронів, особливо в умовах автономних місій.

Сьогодні концепція WPT викликає великий інтерес у транспортному секторі. Бездротова передача енергії почалася з впровадження в портативну електроніку [38] і в першу чергу використовується в смартфонах, військових пристроях і медичному обладнанні, що ілюструється на рисунку 1.6.

Рисунок 1.6 – Застосування бездротової передачі енергії в різних секторах [35]

Вивчення бездротової передачі енергії поділяється на дві категорії залежно від дальності передачі енергії. Для ефективної передачі енергії на відстань менше одного метра використовуються технології передачі ближнього поля [39], такі як ємнісна передачу потужності (CPT, Capacitive Power Transfer), індуктивна передачу потужності (IPT, Inductive Power Transfer) і магнітно-резонансний зв'язок (MRC, Magnetic Resonant Coupling). Для передачі енергії на великі відстані

використовуються технології далекого радіусу дії, такі як лазерна зарядка і мікрохвильова передача енергії (MPT, Microwave Power Transfer). Характеристики основних технологій WPT наведено в таблиці 1.1.

Таблиця 1.1 – Характеристики технологій бездротової передачі енергії

Тип зарядки	Частота	Передана потужність	Відстань передачі
Ємнісна зарядка	4.2 МГц	3.7 Вт	0,13 мм
Індуктивна зарядка	22 кГц	100 кВт	127 мм
Магнітно-резонансна зарядка	60 кГц	818 кВт	50 мм
Лазерне випромінювання	2,4 ГГц	1 Вт	3.66 м
Мікрохвильова передача	5,8 ГГц	50 Вт	–
	2,45 ГГц	4.39 кВт	

У CPT дві паралельні пластини розділені невеликим діелектричним середовищем для протікання електричного поля. Перевагою CPT є передача енергії через металеві бар'єри і низькі втрати потужності в оточуючому металі, що робить його придатним для застосувань з меншими потужностями.

Технологія IPT базується на принципі вільнозв'язаного передавача, де індукція магнітного поля забезпечує передачу енергії між котушками. Система складається з передавальної та приймальної котушок з заземлюючими перетворювачами по обидва боки котушок. Перевагою IPT є зручна експлуатація, безпечність і простота реалізації. IPT зазвичай працює на частотах у діапазоні кілогерців (кГц).

MRC є вдосконаленою формою IPT, яка забезпечує передачу енергії на частотах у межах мегагерців (МГц) із меншими втратами. У цій технології котушки передавача і приймача резонують на одній частоті. Для більшої ефективності між двома котушками розміщується проміжна котушка. Значною перевагою MRC є можливість одночасного заряджання кількох споживачів, навіть якщо вони працюють на різних частотах. Таким чином, MRC ідеально підходить для

одночасного заряджання декількох дронів, які діють скоординовано для досягнення колективної мети.

1.3 Огляд підходів до планування та моніторингу руху групи БПЛА

Системи планування та моніторингу руху групи БПЛА є важливим напрямком сучасних досліджень у сфері робототехніки та автоматизації. Відомі системи планування маршруту здебільшого ґрунтуються на алгоритмах оптимізації, які дозволяють визначати найкоротші або найефективніші шляхи між заданими точками з урахуванням перешкод та обмежень. Однак, у випадку групових польотів, планування ускладнюється через необхідність координації дій кількох апаратів для уникнення зіткнень та забезпечення рівномірного розподілу завдань. Для таких задач застосовуються методи теорії графів, що дозволяють моделювати взаємодію між БПЛА, а також методи теорії транспортних систем, зокрема розв’язання задачі кількох комівояжерів, які допомагають оптимізувати маршрут для групи. Математичні оптимізаційні моделі, зокрема еволюційні алгоритми, можуть бути використані для пошуку оптимальних рішень у складних динамічних умовах, а методи ройового інтелекту (PSO, Ant Colony Optimization) [40] забезпечують ефективну координацію руху групи БПЛА, моделюючи поведінку колективних агентів. У реальних умовах часто використовуються централізовані системи керування, які дозволяють координувати рух і адаптувати маршрути в реальному часі, враховуючи зміни в середовищі та взаємодію між апаратами.

Сучасні системи моніторингу руху БПЛА активно використовують телеметрійні модулі, які передають інформацію про поточний стан кожного апарата до наземної станції. Для цього застосовуються протоколи зв’язку MAVLink та MQTT [41], що дозволяють у реальному часі отримувати дані про координати, швидкість, рівень заряду батареї та працездатність ключових компонентів. Деякі системи, наприклад, Mission Planner або QGroundControl, забезпечують не тільки

моніторинг, але й інтерактивне планування польотів та дистанційне керування групами БПЛА.

У системах групового керування важливим аспектом є побудова моделей взаємодії між апаратами. Централізовані системи використовують один головний БПЛА або наземний сервер для прийняття рішень, тоді як децентралізовані підходи передбачають автономне прийняття рішень кожним апаратом з урахуванням інформації від інших учасників групи. Останній підхід дозволяє підвищити стійкість системи до відмов окремих апаратів, однак потребує складніших алгоритмів обміну даними та оптимізації маршрутів.

Також активно досліджуються методи адаптивного планування, що враховують змінні параметри польоту, такі як рівень заряду батареї, погодні умови чи наявність перешкод. Використання методів машинного навчання дозволяє системам самостійно налаштовувати траєкторії руху та оптимізувати розподіл завдань між БПЛА. Наприклад, алгоритми підкріпленого навчання застосовуються для формування динамічних маршрутів в умовах невизначеності.

Аналіз існуючих систем показує, що більшість із них орієнтовані на окремі аспекти планування або моніторингу, не забезпечуючи комплексного підходу. Це створює потребу у розробленні інтегрованих систем, які поєднуватимуть аналіз параметрів компонентів, планування маршрутів та моніторинг у реальному часі для підвищення ефективності групових польотів.

1.4 Основні виклики та технічні обмеження у плануванні й моніторингу групових польотів БПЛА

Планування та моніторинг руху групи БПЛА є складним завданням, що вимагає врахування численних параметрів, які впливають на ефективність і безпеку виконання польотних місій. Однією з головних проблем є оптимізація маршрутів для групи БПЛА з урахуванням індивідуальних параметрів кожного апарата, таких як рівень заряду батареї, вантажопідйомність, швидкість і дальність польоту. Вибір

оптимальних траєкторій у реальному часі стає ще складнішим у випадках, коли параметри змінюються під час виконання місії.

Ще однією важливою проблемою є координація руху кількох БПЛА для уникнення зіткнень та забезпечення рівномірного розподілу завдань. Для цього необхідно враховувати не лише поточне місцезнаходження апаратів, а й прогнозувати їхні майбутні траєкторії. Недостатньо точні алгоритми координації можуть призвести до перевантаження одних апаратів і неефективного використання інших, що знижує загальну ефективність системи.

Моніторинг стану компонентів БПЛА також є критичною проблемою, оскільки вихід з ладу окремих систем може призвести до втрати апарата або невиконання місії. Наприклад, зниження рівня заряду батареї, перегрів двигунів або несправність сенсорів можуть суттєво вплинути на можливість апарата виконувати поставлені завдання. Важливою задачею є створення системи, яка здатна автоматично аналізувати ці параметри та приймати рішення про зміну маршруту або заміну апарата на резервний.

Ще однією проблемою є передача даних між БПЛА та наземною станцією або між самими апаратами [42]. В умовах великої відстані, перешкод або ворожого втручання можуть виникати затримки або втрати даних, що ускладнює процес планування та моніторингу. Надійні протоколи зв’язку, такі як MAVLink або LoRa, можуть покращити передачу даних, але потребують додаткових ресурсів та енергії, що обмежує автономність БПЛА.

Також важливим аспектом є адаптація системи до змінних зовнішніх умов, таких як вітер, дощ або інші погодні явища. Використання стандартних алгоритмів, які не враховують таких факторів, може призвести до помилок у плануванні маршруту або втрати апаратів. Для вирішення цієї проблеми необхідно розробити адаптивні алгоритми, які враховують реальні параметри польоту та здатні змінювати маршрути у процесі виконання місії.

Комплексне вирішення цих проблем дозволить створити ефективні системи планування та моніторингу групи БПЛА, які забезпечать високу надійність та

продуктивність при виконанні завдань у різних сферах, від військових операцій до цивільного використання.

1.5 Сучасні підходи до управління та моніторингу руху групи БПЛА

Для забезпечення стабільної роботи системи необхідно розглянути декілька підходів, що дозволяють адаптивно керувати польотом, враховувати параметри стану безпілотників та забезпечувати високий рівень координації між ними.

Одним із ключових підходів є використання багаторівневих систем управління, що поєднують централізовані та децентралізовані методи. На верхньому рівні управління приймаються глобальні рішення щодо маршруту та розподілу завдань, тоді як на локальному рівні кожен дрон виконує корекцію власної траєкторії відповідно до змін середовища та стану своїх компонентів. Такий підхід забезпечує як високу адаптивність, так і ефективне використання ресурсів.

Для прийняття оптимальних рішень щодо маршруту можуть застосовуватися методи багатокритеріальної оптимізації. Врахування факторів, таких як залишковий заряд батареї, можливі перешкоди на шляху, погодні умови та розподіл навантаження між БПЛА, дозволяє знаходити рішення, що балансують між мінімізацією енергоспоживання та виконанням завдань у задані терміни. Одним із підходів є використання алгоритмів Парето-оптимізації, які дозволяють визначати набір рішень, що є найкращими за кількома критеріями одночасно.

Ще одним перспективним напрямком є використання цифрових двійників (Digital Twins) [43] у процесі моніторингу та планування польотів. Цифровий двійник дозволяє створювати віртуальну модель кожного дрона та його компонентів, оновлюючи її в реальному часі на основі отриманих телеметричних даних. Це дає змогу точніше контролювати стан БПЛА з меншим часом затримки, забезпечуючи його безпеку та здійснювати коригування маршруту на основі симуляційних розрахунків.

Цифровий двійник вперше був запропонований професором Грівзом з Мічиганського університету. Грівз пояснив, що цифровий двійник складається з фізичного простору, віртуального простору та зв'язку між ними. Основною особливістю цифрового двійника є моделювання фізичного простору у віртуальному просторі з максимальною точністю для покращення взаємодії.

Цифровий двійник дозволяє здійснювати моніторинг реальних об'єктів, що існують у фізичному просторі, у режимі реального часу у віртуальному просторі. Також у віртуальному просторі цифровий двійник може імітувати різні ситуації для конкретного об'єкта, прогнозуючи можливі несправності та деградацію продуктивності. Об'єкти у фізичному просторі можуть бути керовані на основі прогнозованих результатів.

Використання цифрового двійника забезпечує більш точний моніторинг і прогнозування, ніж існуючі наземні станції управління (Ground Control System) [44], завдяки змодельованому віртуальному простору. Крім того, автоматизоване дистанційне управління може виконуватися на основі даних про БПЛА та навколишнє середовище в режимі реального часу.

Наземні станції управління (GCP) – це програмно-апаратний комплекс для керування БПЛА. Вона включає наземну станцію управління, яка дозволяє оператору контролювати дрон у реальному часі, планувати місії, отримувати телеметрію та управляти польотом автономно або вручну.

Апаратна частина включає наземну станцію управління, яка може бути представлена у вигляді ноутбука, планшета або спеціалізованого пульта. Також сюди входить комунікаційне обладнання, зокрема антени, приймачі та передавачі для зв'язку з БПЛА через радіоканал, Wi-Fi, 4G/5G або супутниковий зв'язок. Для забезпечення роботи системи використовуються різні джерела живлення: акумулятори, генератори або мережеве підключення.

Програмна частина включає інтерфейс керування – спеціальне програмне забезпечення (наприклад, Mission Planner, QGroundControl, UgCS), яке дозволяє оператору керувати дроном, задавати маршрут та отримувати телеметричні дані. Також до програмної частини відноситься система телеметрії, що відповідає за

отримання та обробку інформації про висоту, швидкість, положення дрона, стан акумулятора тощо. Деякі GCS мають вбудовані системи автономного керування, які дозволяють дрону виконувати місію без постійного втручання оператора.

Існує кілька видів GCS залежно від умов використання, зокрема:

– Стаціонарні – це великі наземні станції, які зазвичай використовуються у військових або промислових застосуваннях. Вони мають потужне обладнання для керування та обробки великого обсягу даних.

– Портативні – це легкі й мобільні пристрої, такі як планшети, ноутбуки або спеціалізовані пульти. Вони використовуються в польових умовах для швидкого та ефективного управління дроном.

– Хмарні – сучасні GCS, які дозволяють керувати БПЛА через інтернет. Вони дають змогу операторам контролювати дрони з будь-якої точки світу, використовуючи веб-інтерфейси та мобільні додатки.

Для підвищення рівня автономності груп БПЛА можуть застосовуватися технології навчання з підкріпленням (Reinforcement Learning), що дозволяють системі самостійно адаптувати маршрути на основі накопиченого досвіду. Наприклад, алгоритми глибокого навчання можуть використовувати історичні дані польотів для визначення найефективніших стратегій обходу перешкод та оптимального розподілу дронів у просторі.

Для зменшення ризиків втрати БПЛА через несприятливі умови, перешкоди на шляху та оптимізації витрат енергії можливо інтегрувати модель прогнозування змін у зовнішньому середовищі. Аналіз цих даних у реальному часі дозволяє адаптувати маршрути з урахуванням напрямку та швидкості вітру, опадів та інших погодних факторів.

1.6 Постановка задачі

Система планування та моніторингу руху групи БПЛА на основі аналізу параметрів їх компонентів має на меті забезпечити ефективне керування групою дронів для виконання різноманітних завдань у динамічних умовах. Для реалізації

такої системи необхідно вирішити комплекс завдань, пов'язаних із вибором оптимальних маршрутів, координацією роботи апаратів та безперервним моніторингом їхнього стану.

Однією з задач є розроблення алгоритмів планування маршрутів, які враховують параметри компонентів кожного БПЛА, зокрема висота, швидкість та дальність польоту (час польоту), рівень заряду батареї, вантажопідйомність,. Це передбачає створення математичних моделей, що дозволяють визначати оптимальні траєкторії з мінімальними витратами енергії та максимальним покриттям зони обслуговування.

Важливою складовою є координація руху групи БПЛА, яка дозволяє розподіляти завдання між апаратами таким чином, щоб уникати зіткнень та перевантаження окремих дронів. Для цього необхідно розробити механізми комунікації між апаратами та синхронізації їхніх дій у реальному часі.

Наступною задачею є створення системи моніторингу стану компонентів БПЛА, яка забезпечує постійний аналіз параметрів, таких як температура двигунів, рівень заряду акумуляторів, працездатність сенсорів та якість зв'язку. На основі цих даних система має приймати рішення про зміну маршруту, передачу завдань іншому апарату або повернення дрона на базу для технічного обслуговування.

Загалом, постановка задачі передбачає створення комплексної системи, яка поєднує функції планування маршрутів, координації роботи групи БПЛА та моніторингу стану компонентів, забезпечуючи ефективне та безпечне виконання завдань у різних умовах експлуатації.

1.7 Висновки до першого розділу

Досліджуючи предметну область проведено аналіз того де і для яких цілей можуть використовуватися БПЛА або вже використовуються. Визначено за якими ознаками їх класифікують і як поділяють на різні типи за призначенням. Визначено та описано з яких компонентів складається типовий БПЛА.

У процесі дослідження було проаналізовано існуючі системи планування та моніторингу руху групи БПЛА, а також визначено основні проблеми, пов'язані з керуванням групою безпілотних літальних апаратів на основі параметрів їх компонентів. Проведений аналіз показав, що сучасні системи часто мають обмежену функціональність у частині врахування індивідуальних характеристик кожного дрона, що може призводити до неефективного використання ресурсів та зниження продуктивності групи в цілому.

Виявлено, що для підвищення ефективності роботи групи БПЛА необхідно застосовувати комплексний підхід, який поєднує оптимізацію маршрутів, координацію руху та моніторинг стану компонентів у реальному часі. Особливу увагу слід приділяти системам передачі даних, оскільки якісний обмін інформацією між апаратами є ключовим фактором для синхронізації їхньої роботи та уникнення конфліктних ситуацій.

Розглянуті проблеми підтвердили необхідність розроблення нових алгоритмів, що дозволяють враховувати параметри компонентів БПЛА при плануванні руху та адаптувати маршрути до змінних зовнішніх умов. Крім того, інтеграція механізмів моніторингу технічного стану апаратів дозволить підвищити надійність системи та мінімізувати ризики втрати дронів під час виконання місій.

Результати дослідження закладають основу для подальшої розробки системи планування та моніторингу руху групи БПЛА, яка сприятиме підвищенню ефективності роботи групи, оптимізації використання енергетичних ресурсів та забезпеченню безпеки виконання завдань у різних умовах експлуатації.

2 АНАЛІЗ АЛГОРИТМІВ ДЛЯ ПЛАНУВАННЯ ТА МОНІТОРИНГУ МАРШРУТІВ РУХУ ГРУПИ БПЛА

2.1 Машинне навчання

У процесі розроблення алгоритмів планування та моніторингу маршрутів руху групи безпілотних літальних апаратів було обґрунтовано доцільність застосування методів машинного навчання як ефективного підходу до автоматизованого прийняття рішень в умовах динамічного середовища та часткової спостережуваності. На відміну від класичних алгоритмів, що вимагають ручного налаштування та жорстко заданих правил, моделі машинного навчання здатні адаптуватися до нових ситуацій, самостійно вдосконалювати власну поведінку на основі накопиченого досвіду та забезпечувати високу якість планування у режимі реального часу.

Машинне навчання – це напрям штучного інтелекту, який передбачає розроблення та використання алгоритмів, здатних автоматично поліпшувати свою продуктивність на основі накопиченого досвіду. Основна мета машинного навчання полягає в побудові моделей, що здатні виявляти закономірності у даних та приймати рішення або робити прогнози без прямого програмування [45].

Машинне навчання є ключовою технологією у багатьох сучасних галузях, включаючи медицину, фінансову аналітику, промислову автоматизацію, робототехніку та транспорт. Особливої актуальності воно набуває у сфері автономного управління безпілотними літальними апаратами, де його застосування забезпечує принципово новий рівень інтелектуальної поведінки систем. Завдяки здатності до аналізу великих обсягів даних у режимі реального часу, машинне навчання дозволяє ефективно вирішувати широкий спектр задач: від локального й глобального планування маршрутів до моніторингу навколишнього середовища, виявлення цільових об’єктів, класифікації ситуацій та адаптивного ухвалення рішень. Використання методів навчання з підкріпленням, глибокого навчання та мультиагентного навчання у поєднанні з даними сенсорів і геопросторовою інформацією дозволяє створювати адаптивні системи керування, здатні до

самонавчання, співпраці та динамічного реагування на зміну умов середовища. Таким чином, машинне навчання є критично важливим компонентом сучасних систем автономного управління БПЛА.

Залежно від типу доступних даних та мети навчання, машинне навчання поділяється на три основні підходи: контрольоване машинне навчання, неконтрольоване навчання та машинне навчання з підкріпленням.

2.1.1 Контрольоване машинне навчання

Контрольоване машинне навчання є одним із базових типів машинного навчання, що передбачає навчання моделі на основі мічених даних, тобто таких, для яких відомі як вхідні характеристики (ознаки), так і відповідні їм вихідні значення (мітки) [46]. Метою цього підходу є побудова функції, яка на основі нових вхідних даних зможе передбачити відповідні вихідні значення з прийнятною точністю. Принцип дії контрольованого машинного навчання представлено на рисунку 2.1.

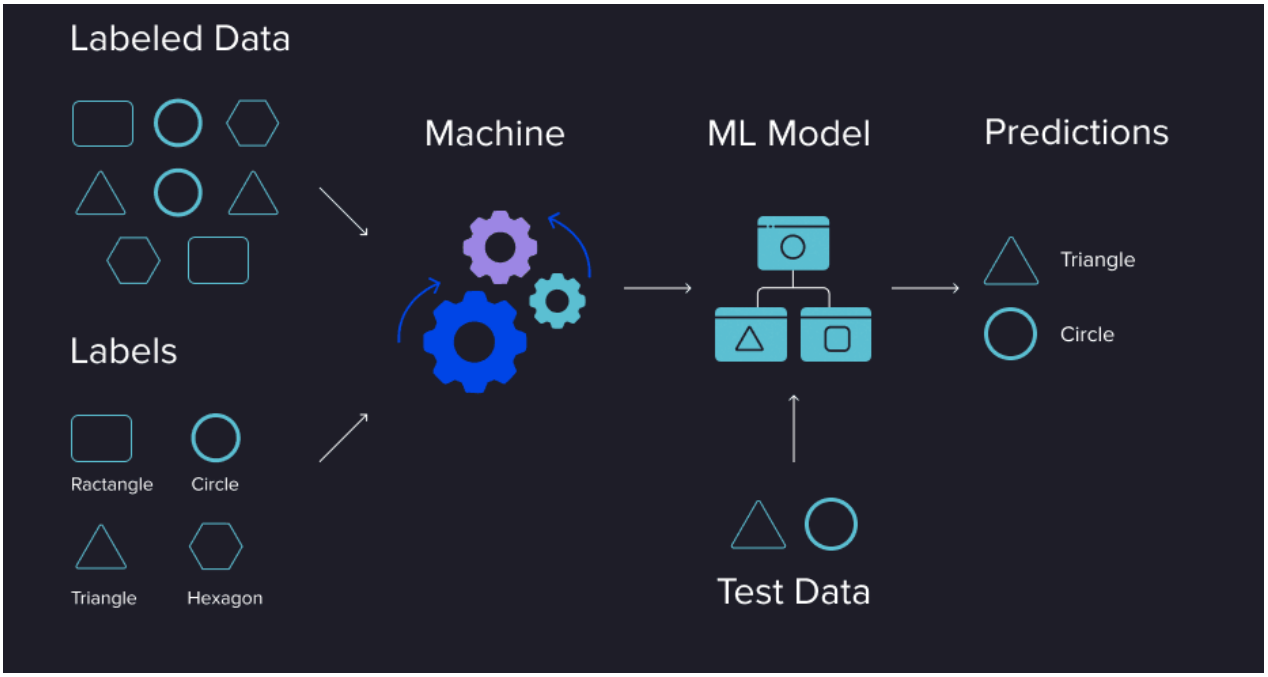


Рисунок 2.1 – Принцип дії контрольованого машинного навчання [47]

Контрольоване навчання застосовується у випадках, коли необхідно здійснити класифікацію об’єктів, прогноз подій або числову оцінку деякого параметра. Основні типи задач, що розв’язуються методами контрольованого навчання, включають класифікацію або регресію.

Задача класифікації полягає у віднесенні об’єкта до одного з наперед визначених дискретних класів на основі його вхідних характеристик (ознак). Алгоритми класифікації формують модель, яка вивчає залежність між ознаками та класами на основі навчального набору даних, а згодом використовує її для передбачення класу нових, невідомих об’єктів. Задачі класифікації широко застосовуються у розпізнаванні образів, виявленні аномалій, аналізі текстів, медичній діагностиці, а також в управлінні БПЛА, наприклад для ідентифікації типу об’єкта в полі зору дрона чи визначення поведінкових шаблонів цілей.

Залежно від кількості класів, задачі класифікації поділяються на бінарні (класифікація на два класи, наприклад «перешкода/не перешкода») та мультикласові (класифікація на три і більше класів). Серед поширених алгоритмів класифікації потрібно виділити такі як логістична регресія, дерева рішень, метод опорних векторів (SVM), метод k найближчих сусідів, наївний баєсівський класифікатор.

Логістична регресія призначена для розв’язання задач класифікації, зокрема бінарної. На відміну від лінійної регресії, її вихід є ймовірністю належності об’єкта до певного класу, після чого застосовується порогове значення для прийняття остаточного рішення [48]. Тому цей алгоритм легко застосовується разом з сигмоїдною функцією, що представлена на рисунку 2.2.

Функція отримує на вхід незалежні змінні та повертає ймовірність у діапазоні від 0 до 1. Наприклад, при класифікації об'єктів на два класи (клас 0 та клас 1), якщо значення логістичної функції перевищує поріг 0.5, об'єкт відносять до класу 1, в іншому випадку – до класу 0.

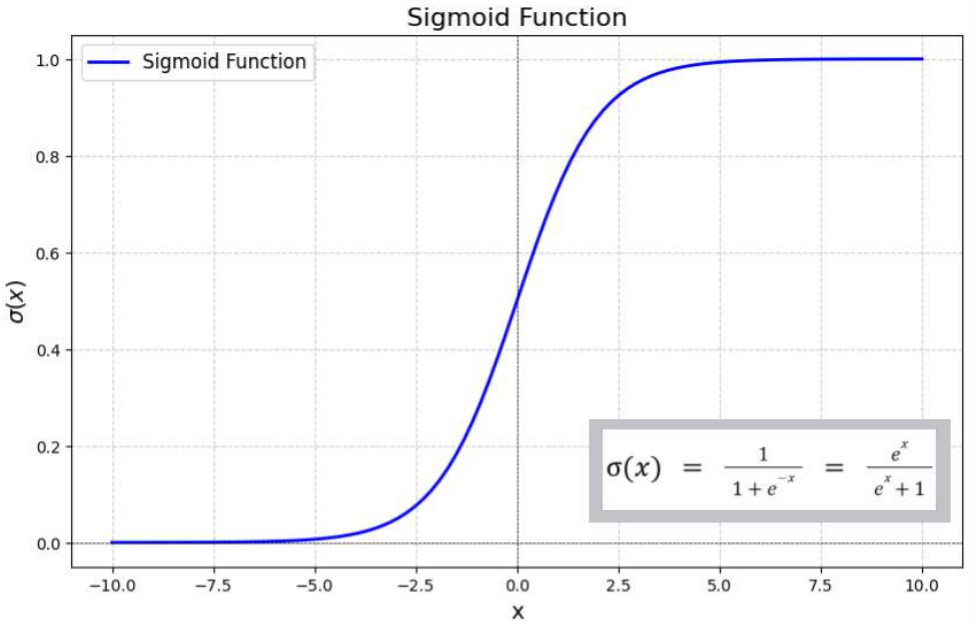


Рисунок 2.2 – Сигмоїдна функція логістичної регресії [49]

Дерева рішень – це алгоритми, що імітують процес прийняття рішень за допомогою послідовного розгалуження. Вони легко інтерпретуються, але можуть бути схильними до перенавчання на складних наборах даних [50]. Структура дерева рішень має чітку ієрархію, від кореневого вузла йде розгалуження на внутрішні вузли і від них на листові вузли. Типовий приклад структури дерева рішень представлено на рисунку 2.3.

Рисунок 2.3 – Приклад структури дерева рішень

Як видно з рисунка 2.3, дерево рішень починається з кореневого вузла, який не має вхідних гілок. Від нього відходять гілки до внутрішніх вузлів, що виконують функцію прийняття рішень. Кореневий та внутрішні вузли аналізують доступні ознаки, поділяючи дані на більш однорідні підгрупи. Кінцеві підгрупи представлені листовими вузлами, вони представляють всі допустимі результати у вибірці.

Метод опорних векторів – це алгоритм, що ефективно виконує класифікацію шляхом пошуку оптимальної гіперплощини, яка максимально відокремлює класи. Підходить для задач з високою розмірністю ознак та складною нелінійною структурою [51]. Класифікацію методом опорних векторів на два класи зображено на рисунку 2.4.

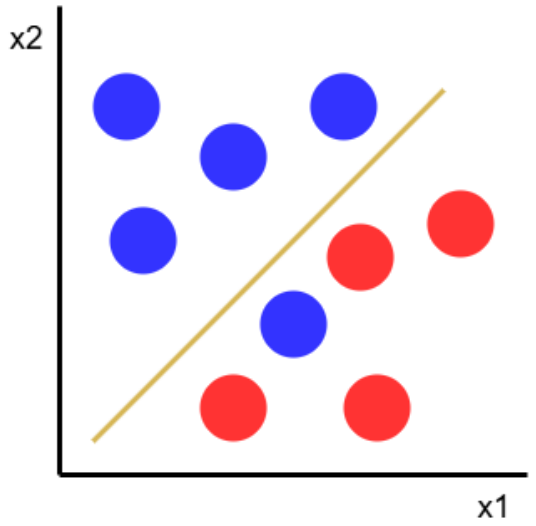


Рисунок 2.4 – Класифікація методом опорних векторів

Метод k найближчих сусідів (k-NN) є простим та інтуїтивно зрозумілим алгоритмом машинного навчання, який використовується переважно для задач класифікації, а також регресії [52]. Суть методу полягає в тому, що новий об’єкт класифікується на основі більшості голосів серед k найближчих (за певною метрикою відстані) об’єктів у навчальній вибірці. Таким чином, передбачається, що об’єкти з подібними ознаками мають належати до одного класу.

Ключовими чинниками, що впливають на ефективність алгоритму, є вибір значення параметра k та метрики відстані (найчастіше використовується евклідова,

мангеттенська або косинусна відстань). Занадто мале значення k може призводити до нестабільних рішень, чутливих до шуму, тоді як занадто велике – до втрати деталізації та зміщення результату.

Метод k-NN не потребує етапу навчання у традиційному розумінні – всі обчислення відбуваються під час класифікації нового об’єкта, що зумовлює високе навантаження на обчислювальні ресурси під час виконання, особливо за великої кількості навчальних даних. Незважаючи на це, k-NN є ефективним у багатьох практичних застосуваннях, таких як розпізнавання образів, рекомендаційні системи, виявлення аномалій та інші задачі, де важлива простота реалізації та інтерпретованість. На рисунку 2.5 представлена робота методу k найближчих сусідів.

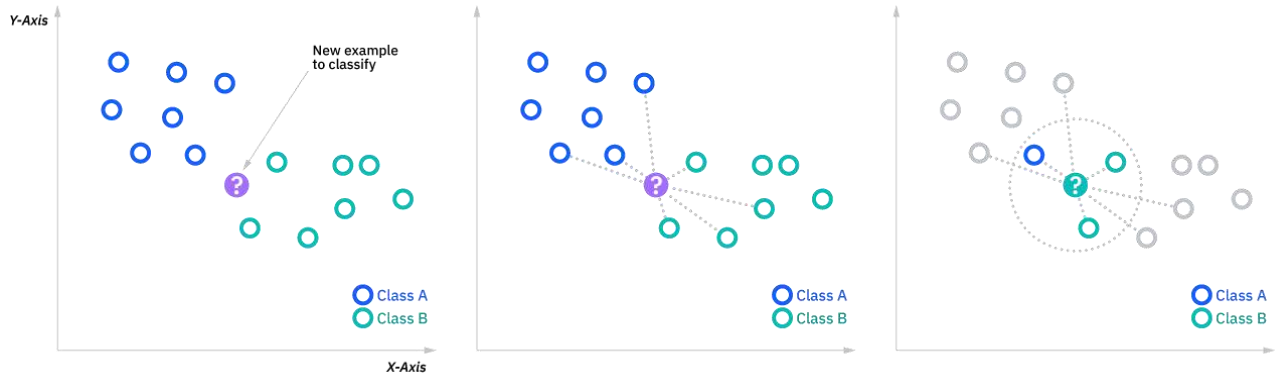


Рисунок 2.5 – Робота методу k найближчих сусідів [53]

Наївний баєсівський класифікатор є ймовірнісним методом класифікації, що ґрунтується на теоремі Байєса та припущенні про незалежність ознак. Незважаючи на свою «наївність», тобто припущення про відсутність кореляції між ознаками, цей алгоритм демонструє високу ефективність у багатьох практичних задачах, особливо за наявності великого обсягу текстових або категоріальних даних [54]. Класична формула теореми Байєса використовується для обчислення ймовірності належності нового об’єкта до кожного з можливих класів і виглядає як:

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)}, \quad (2.1)$$

де $P(A|B)$ – це ймовірність того, що об’єкт належить до класу А за умови, що відомі його ознаки В;

$P(B|A)$ – ймовірність спостерігати набір ознак В, якщо об’єкт належить класу А;

$P(A)$ – апріорна ймовірність того, що об’єкт належить до класу А;

$P(B)$ – ймовірність спостерігати набір ознак В незалежно від того, до якого класу належить об’єкт.

Кінцеве рішення приймається на користь того класу, для якого ймовірність найбільша. Завдяки своїй простоті, високій швидкості роботи та невибагливості до обсягу ресурсів, наївний баєсівський класифікатор широко використовується у задачах фільтрації спаму, класифікації документів, розпізнавання мовлення та інших застосуваннях у режимі реального часу. На рисунку 2.6 відображено логіку роботи наївного баєсівського класифікатора.

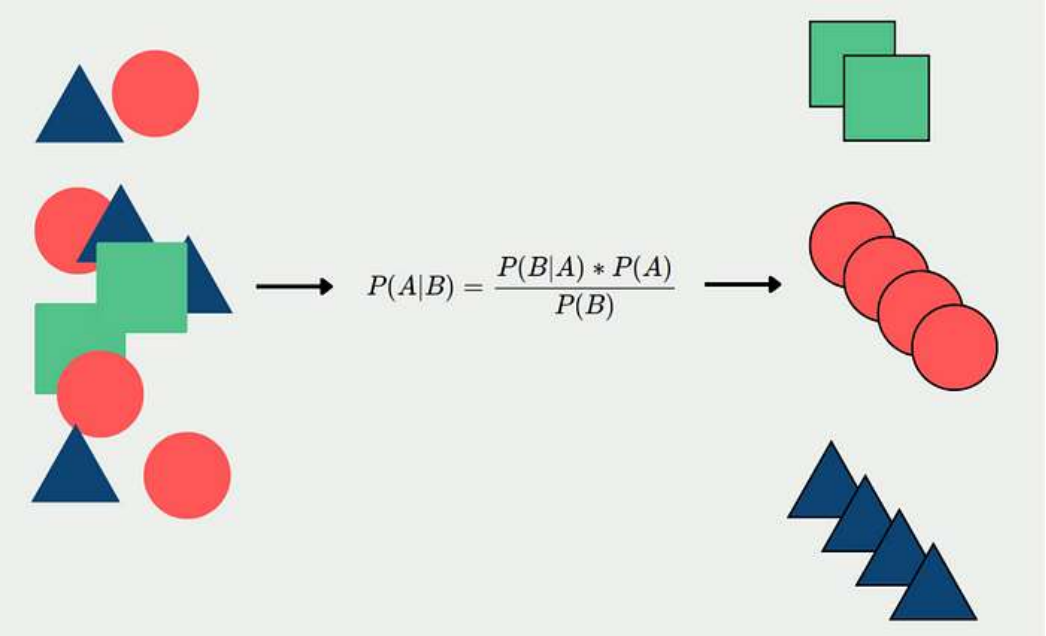


Рисунок 2.6 – Логіка роботи наївного баєсівського класифікатора [55]

У задачах регресії модель передбачає неперервне числове значення на основі вхідних даних. Метою є знаходження функціональної залежності між набором вхідних ознак та числовою цільовою змінною. Регресійні моделі використовуються тоді, коли необхідно оцінити величину певного параметра, наприклад: прогноз температури, швидкості, часу, координат, рівня заряду тощо. У контексті застосування до БПЛА, задачі регресії можуть охоплювати прогнозування відстані до об'єкта, часу досягнення цілі, обсягів спожитої енергії або інших показників, важливих для автономного керування.

Основними алгоритмами регресії є лінійна регресія, поліноміальна регресія, методи ансамблевого навчання (наприклад, градієнтне підсилення або випадковий ліс), нейронні мережі та інші моделі, здатні працювати з числовими результатами.

Лінійна регресія – це базовий алгоритм регресійного аналізу, що встановлює лінійну залежність між незалежними змінними та цільовою змінною. Модель має аналітичний вигляд, що спрощує її інтерпретацію. Використовується для прогнозування числових значень у випадках, коли зв’язок між змінними є приблизно лінійним [56]. Як показано на рисунку 2.7, модель лінійної регресії дозволяє передбачити значення залежної змінної Y на основі значень незалежної змінної X, враховуючи похибку моделі.

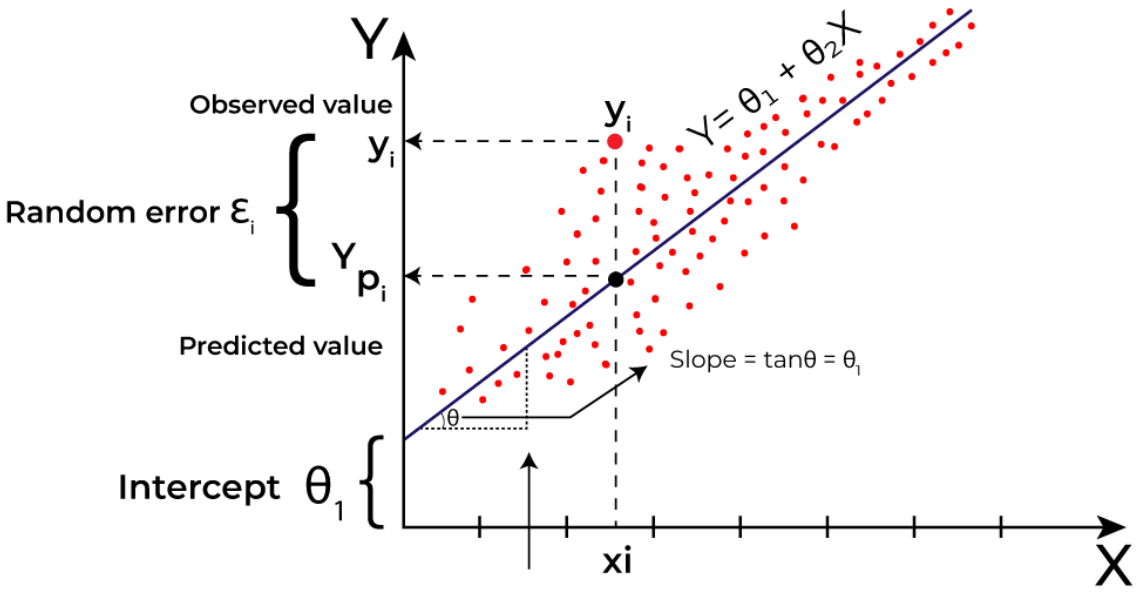


Рисунок 2.7 – Схематичне зображення моделі лінійної регресії [57]

Математичне представлення лінійної регресії подається у вигляді наступного рівняння:

$$y = \beta_0 + \beta_1 x, \tag{2.2}$$

де y – це прогнозоване значення (залежна змінна);

β_0 – точка перетину з віссю у (значення у, коли х = 0);

β_1 – нахил прямої (наскільки змінюється у при зміні х);

x – вхідні дані (незалежна змінна).

Поліноміальна регресія є узагальненням лінійної регресії, яка дозволяє моделювати нелінійні залежності між незалежними змінними та цільовою змінною [58]. У цьому методі зв’язок між змінними описується як поліном певного ступеня, що дає змогу враховувати складніші закономірності в даних. Поліноміальна регресія добре працює в умовах, коли є потреба апроксимувати криволінійну траєкторію, прогнозувати нелінійні тренди або аналізувати динамічні процеси. Водночас, зростання ступеня полінома збільшує ризик перенавчання, тому вибір моделі потребує обґрунтованої балансування між точністю й узагальнюваністю.

Математичне представлення поліноміальної регресії подається у вигляді наступного рівняння:

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + ... + \beta_n x^n + \epsilon, \tag{2.3}$$

де y – це залежна змінна;

x – незалежна змінна;

$\beta_0, \beta_1, ..., \beta_n$ – коефіцієнти поліноміальних членів;

n – степінь полінома;

ϵ – представляє член помилки.

На рисунку 2.8 показано приклад апроксимації даних за допомогою поліноміальної регресії, яка краще узгоджується з нелінійною залежністю між змінними порівняно з лінійною моделлю.

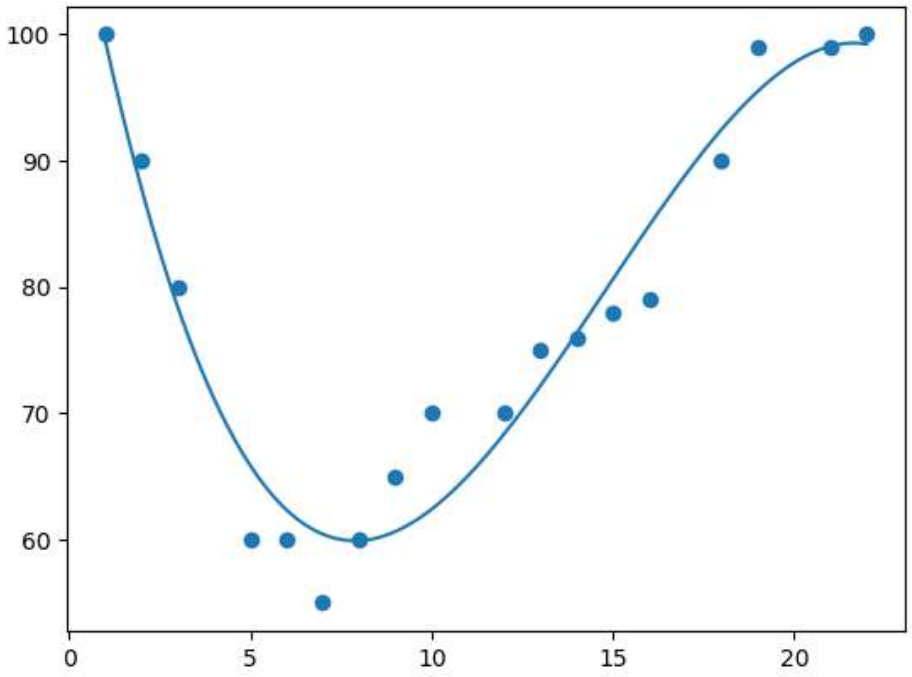


Рисунок 2.8 – Апроксимації даних за допомогою поліноміальної регресії [59]

Градієнтне підсилення – це потужний ансамблевий метод машинного навчання, який створює композитну (сильну) модель шляхом поетапного додавання слабких моделей, найчастіше – дерев рішень з обмеженою глибиною. Кожна наступна модель навчається на залишкових помилках попередньої, тобто на тих прикладах, які були класифіковані або передбачені неправильно. Таким чином, нові моделі поступово зменшують похибку всієї системи, що призводить до високої точності кінцевої моделі.

Популярні реалізації цього методу включають алгоритми XGBoost, LightGBM, CatBoost.

Процес навчання в градієнтному підсиленні реалізується як покрокова оптимізація певної функції втрат за допомогою градієнтного спуску, що й зумовило назву методу. Гнучкість у виборі функцій втрат дозволяє застосовувати градієнтне

підсилення як до задач регресії, так і до класифікації [60]. На рисунку 2.9 зображено загальний принцип роботи градієнтного підсилення.

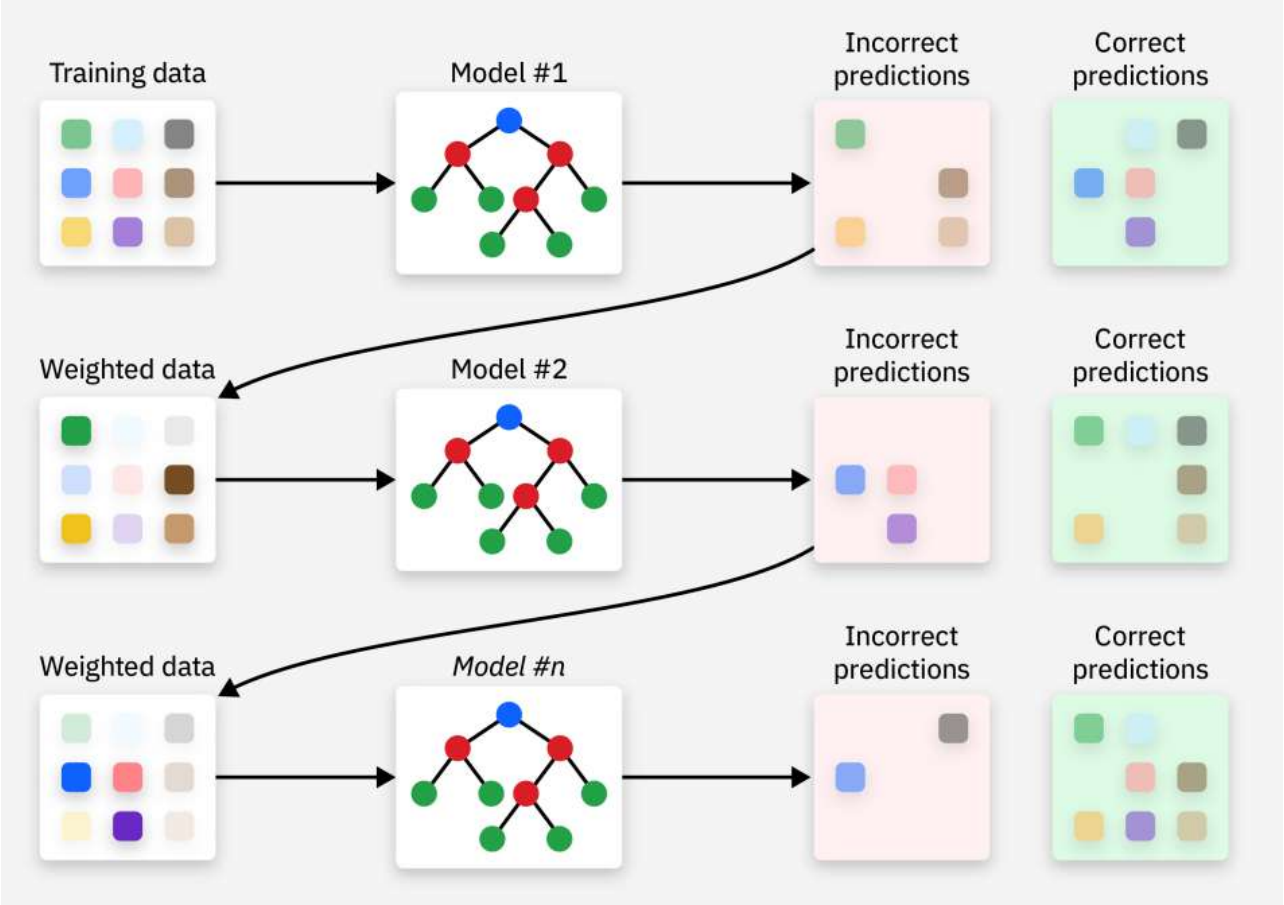


Рисунок 2.9 – Робота алгоритму градієнтного підсилення [61]

Метод випадкового лісу (Random Forest) – це ансамблевий алгоритм машинного навчання, що ґрунтується на об'єднанні великої кількості незалежно побудованих дерев рішень, кожне з яких навчається на випадковій підмножині вихідних даних. Такий підхід реалізує принцип беггінгу (Bootstrap Aggregating), який полягає у зниженні дисперсії (варіативності) моделі шляхом усереднення результатів кількох слабких моделей, побудованих на змінних вибірках.

У процесі класифікації кожне дерево формує власне рішення, а фінальний прогноз визначається голосуванням більшості (у задачах класифікації) або середнім значенням (у задачах регресії) [62]. На рисунку 2.10 показано роботу методу випадкового лісу. Крім того, при побудові кожного дерева відбір ознак для

поділу вузлів також здійснюється випадковим чином, що сприяє зменшенню кореляції між деревами та покращенню узагальнюваності моделі.

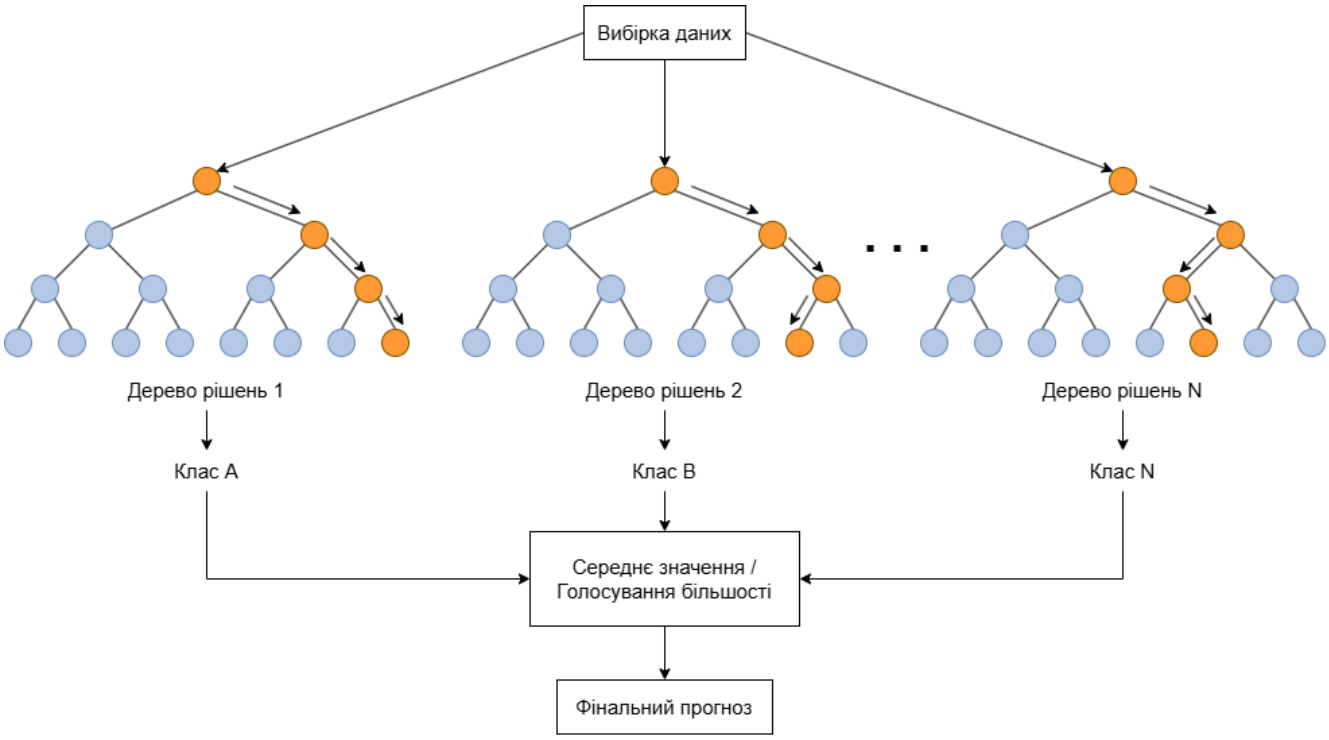


Рисунок 2.10 – Робота методу випадкового лісу

2.1.2 Неконтрольоване машинне навчання

Наступним підходом машинне навчання є неконтрольоване – це тип машинного навчання, у якому моделі навчаються на основі немічених даних, тобто без наявності заздалегідь визначених цільових значень. Метою цього підходу є виявлення прихованої структури, закономірностей або взаємозв'язків у даних без зовнішнього втручання або нагляду.

На відміну від контрольованого навчання, яке орієнтоване на передбачення певної мітки, неконтрольоване навчання зосереджене на групуванні даних, виявленні аномалій, зменшенні розмірності або виявленні латентних (прихованих) змінних. Основними задачами, які вирішуються в межах неконтрольованого навчання, є кластеризація, асоціативне навчання та зниження розмірності [63]. На рисунку 2.11 зображено процес навчання без учителя, у якому модель самостійно

аналізує необроблені вхідні дані та групує їх за схожістю без наявності попередньо маркованого навчального набору.

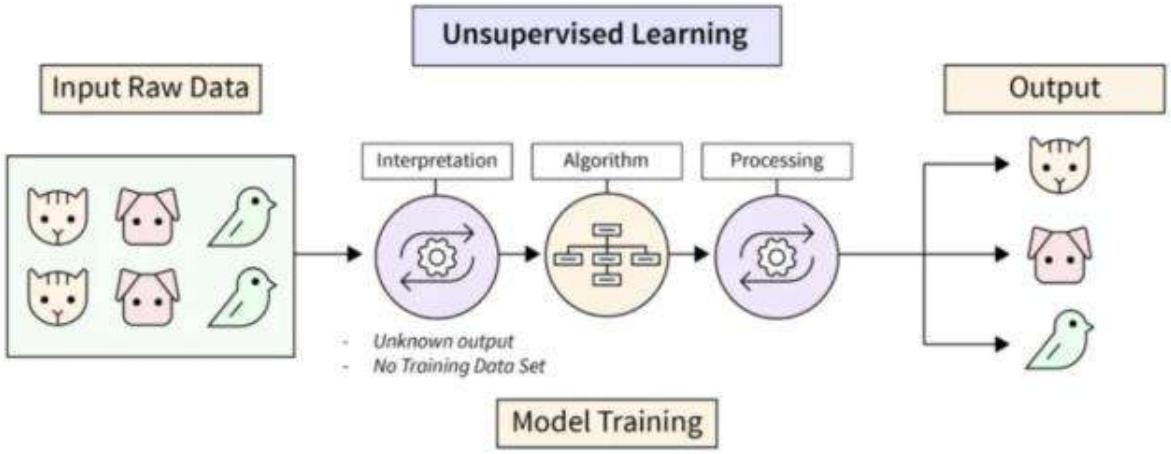


Рисунок 2.11 – Схема неконтрольованого навчання [64]

Кластеризація є ключовим методом неконтрольованого машинного навчання, який передбачає поділ набору даних на групи (кластери) таким чином, щоб об’єкти всередині одного кластера мали високий ступінь подібності між собою, а об’єкти з різних кластерів – суттєво відрізнялися. Цей підхід дає змогу автоматично виявляти структуру в даних, що особливо корисно при відсутності мічених прикладів або апріорного знання про розподіл класів.

Найпоширенішим і водночас базовим алгоритмом кластеризації є метод k-середніх (k-Means) [65]. Його суть полягає в ітеративному групуванні об’єктів навколо k заздалегідь визначених центрів кластерів (центроїдів). На кожній ітерації алгоритм спочатку призначає кожен об'єкт до найближчого центроїда за обраною метрикою (зазвичай використовується евклідова відстань), а потім обчислює нові координати центрів як середнє арифметичне об'єктів, що належать до відповідного кластера. Процес повторюється доти, поки центроїди не перестануть змінюватися або буде досягнуто заданої кількості ітерацій.

Метод k-середніх є ефективним та обчислювально недорогим, що робить його придатним для роботи з великими обсягами даних. Проте він має низку обмежень: вимагає попереднього визначення кількості кластерів k, є чутливим до

початкової ініціалізації центроїдів і може давати неоднозначні результати при наявності кластерів різної форми, розміру або щільності. Для подолання цих недоліків застосовуються модифікації, такі як k-Means++ для покращеного вибору початкових центрів, або інші алгоритми кластеризації – зокрема DBSCAN чи ієрархічна кластеризація, які не потребують заздалегідь заданого числа кластерів і краще працюють у складних просторових структурах. Кластеризацію за допомогою методу k-середніх показано на рисунку 2.12.

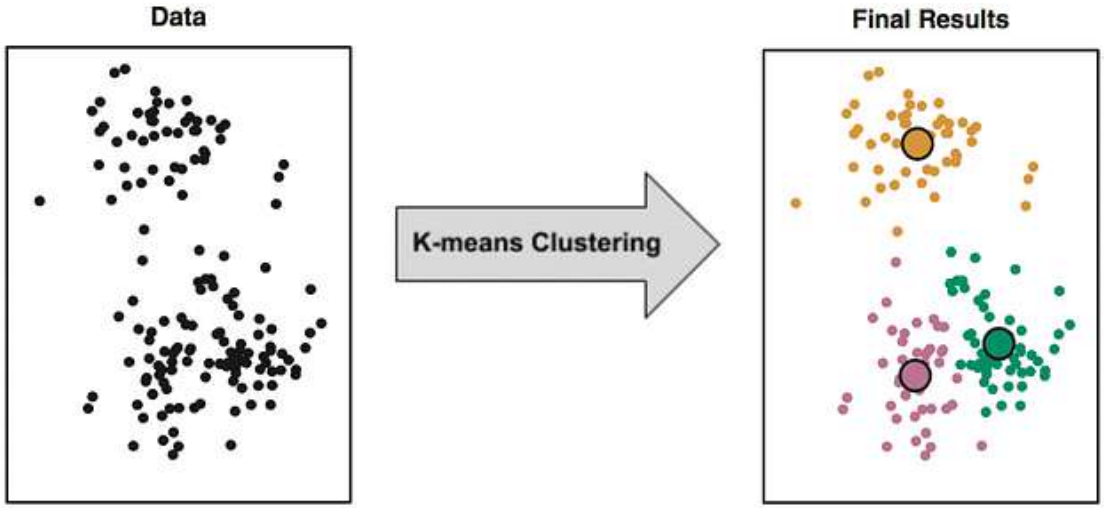


Рисунок 2.12 – Робота методу k-середніх [66]

Асоціативне навчання (або виявлення асоціативних правил) – це підхід у неконтрольованому машинному навчанні, що спрямований на виявлення складних логічних взаємозв’язків і залежностей між змінними у великих масивах даних. Основна мета полягає у формулюванні правил вигляду «якщо..., то...», які виявляють ймовірнісні закономірності у спільній появі певних ознак або подій. Наприклад, правило у роздрібній торгівлі, зокрема для аналізу кошика покупця, «якщо клієнт купив хліб, то з високою ймовірністю придбає масло» є типовим результатом такого навчання.

Найпоширенішим алгоритмом у цій категорії є алгоритм Apriori [67], який здійснює пошук частих наборів ознак (itemsets), підмножин ознак, що часто трапляються разом, із наступною генерацією асоціативних правил на їх основі. Apriori використовує принцип монотонності: якщо деякий набір ознак є нечастим,

то всі його надмножини також не є частими, що дозволяє ефективно зменшувати обсяг пошуку. Для кожного з отриманих правил обчислюються такі характеристики, як: підтримка, достовірність та підняття.

Зниження розмірності – це процес перетворення багатовимірного простору ознак у простір меншої розмірності за збереження максимальної кількості суттєвої інформації [68]. Такий підхід дозволяє вирішити низку важливих задач, серед яких: зменшення обчислювальної складності, спрощення моделей, усунення надлишкових або корельованих ознак, а також візуалізація високорозмірних даних у 2D або 3D-просторі для інтерпретації.

Однією з основних причин застосування методів зниження розмірності є подолання так званого «прокляття розмірності», коли зростання кількості ознак призводить до експоненційного збільшення об'єму простору.

Класичним методом зниження розмірності є метод головних компонент (Principal Component Analysis, PCA) [69]. Він здійснює лінійне перетворення вхідних даних у нову ортогональну систему координат так, щоб перша головна компонента мала найбільшу дисперсію, друга – наступну за значущістю тощо. Це дозволяє виділити основні напрями варіації в даних і відкинути другорядні. Метод є обчислювально ефективним, добре масштабується та використовується для попередньої обробки даних перед побудовою моделей машинного навчання. Приклад з перетворенням вхідних даних методом головних компонент представлено на рисунку 2.13.

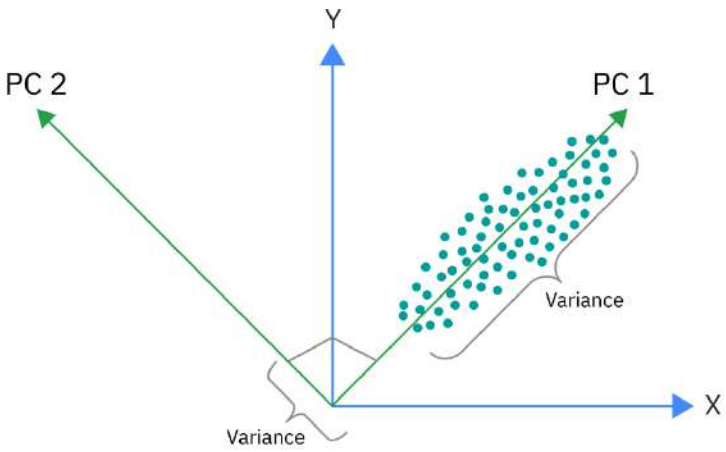


Рисунок 2.13 – Робота методу головних компонент [70]

2.1.3 Машинне навчання з підкріпленням

Машинне навчання з підкріпленням є окремим напрямом машинного навчання, що ґрунтується на принципі взаємодії агента із середовищем з метою максимізації сукупної винагороди за певний період часу [71]. Цей підхід не потребує мічених даних, натомість агент навчається на основі зворотного зв'язку від середовища – позитивної або негативної винагороди, що залежить від обраних дій.

У класичному формулюванні процес навчання описується за допомогою марковського процесу прийняття рішень, який визначається множиною станів, множиною дій, функцією винагороди, функцією переходу та політикою. Мета агента полягає у знаходженні оптимальної стратегії (політики) дій, яка забезпечує максимальне очікуване значення сукупної винагороди в довгостроковій перспективі. На рисунку 2.14 зображена схема машинного навчання з підкріпленням. Залежно від наявності або відсутності моделі середовища, розрізняють два основні підходи: навчання з використанням моделі і безмодельне навчання.

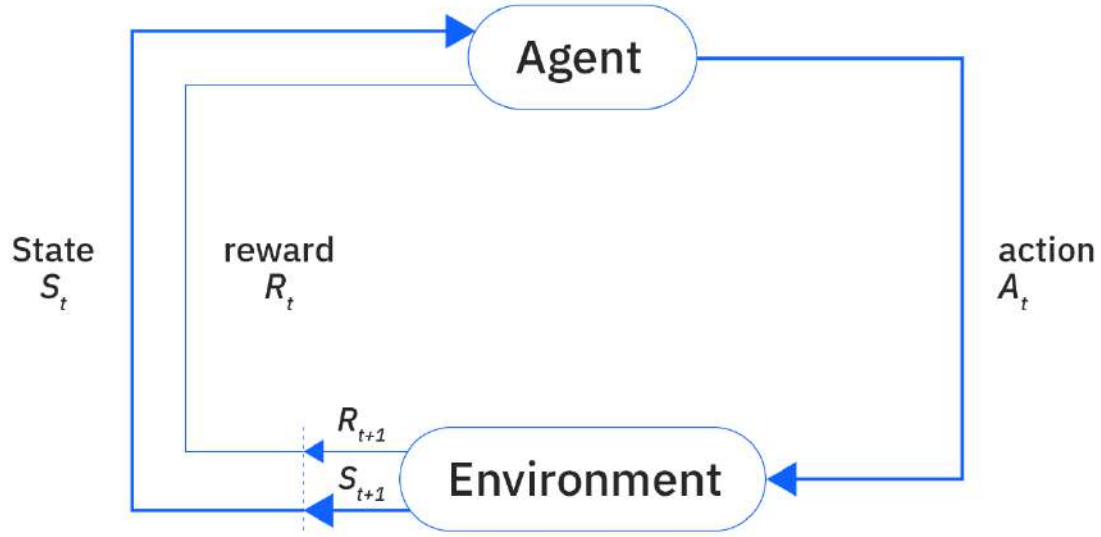


Рисунок 2.14 – Схема машинного навчання з підкріпленням [72]

Навчання з використанням моделі – це підхід у навчанні з підкріпленням, який передбачає, що агент має або формує власну модель середовища, тобто моделює динаміку переходів між станами та очікувану винагороду внаслідок виконання певної дії. Така модель дозволяє агенту планувати послідовність дій, прогнозуючи можливі наслідки без необхідності здійснювати реальні дії в середовищі, що значно прискорює процес навчання та знижує ризики.

У цьому підході агент навчається у два етапи. Спочатку будується динамічна модель середовища, яка апроксимує ймовірні переходи між станами на основі накопиченого досвіду. Потім ця модель використовується для планування оптимальних дій за допомогою методів динамічного програмування або симуляції.

Безмодельне навчання з підкріпленням – це підхід, за якого агент навчається ухвалювати оптимальні рішення без явного уявлення про динаміку середовища. Інакше кажучи, агент не будує внутрішню модель переходів між станами чи функцію винагороди, а опановує поведінку виключно на основі власного досвіду взаємодії із середовищем. Такий підхід є особливо цінним у випадках, коли середовище є складним, нелінійним або частково спостережуваним, і побудова точної моделі є технічно складною або неможливою.

У межах безмодельного навчання з підкріпленням розрізняють дві основні групи методів, що засновані на функції цінності або на політиці. Методи, засновані на функції цінності навчають функцію цінності (наприклад, Q-функцію), яка оцінює корисність дії у певному стані. Методи, засновані на політиці безпосередньо навчають політику агента (тобто ймовірнісний розподіл дій), не оцінюючи цінність кожного стану. Акторно-критичні методи (Actor-Critic) комбінують переваги обох підходів: актор (мережа політик) визначає дію, а критик (мережа цінностей) оцінює її якість.

До поширених безмодельних алгоритмів належить метод Q-Learning, SARSA та пошук по дереву Монте-Карло.

Q-навчання – це алгоритм який використовується для навчання агентів приймати оптимальні рішення шляхом взаємодії з середовищем [73]. Він допомагає агенту досліджувати різні дії та дізнаватися, які з них призводять до кращих

результатів. Агент використовує метод спроб і помилок, щоб визначити, які дії призводять до винагород (хороші результати) або штрафів (погані результати).

З часом він покращує своє прийняття рішень, оновлюючи Q-таблицю, яка зберігає Q-значення, що представляють очікувані винагороди за виконання певних дій у заданих станах. Агент переходить через різні стани, виконуючи дії та отримуючи винагороду. Процес триває, доки агент не досягне термінального стану, що завершує епізод. Агент оновлює Q-значення за формулою:

$$Q(S,A) \leftarrow Q(S,A) + \alpha\big(R + \gamma Q(S',A') - Q(S,A)\big), \tag{2.4}$$

де S – це поточний стан;

A – дія, яку виконує агент;

S' – наступний стан, до якого переходить агент;

A' – найкраща наступна дія у стані S';

R – винагорода, отримана за виконання дії A у стані S;

γ – коефіцієнт дисконтування , який збалансовує негайні винагороди з майбутніми винагородами;

α – швидкість навчання , яка визначає, наскільки нова інформація впливає на старі значення Q.

SARSA – алгоритм який є on-policy, тобто навчає політику, яка відповідає тим діям, які реально виконує агент під час навчання, на відміну від Q-Learning, який є off-policy методом і оновлює функцію цінності, виходячи з оптимальних дій у майбутньому стані. Основна ідея SARSA полягає в оновленні Q-значення для кожної пари стан-дія на основі фактичного досвіду (тобто того, що робить агент, дотримуючись своєї політики). Q-значення оновлюється за допомогою наступного рівняння Беллмана для SARSA [74]:

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha\big[r_{t+1} + \gamma Q(s_{t+1},a_{t+1}) - Q(s_t,a_t)\big], \tag{2.5}$$

де $Q(s_t, a_t)$ – це поточне значення Q для пари стан-дія на кроці часу t;

α – швидкість навчання (значення від 0 до 1), яка визначає, наскільки оновлюються Q-значення;

r_{t+1} – винагорода, отримана після виконання дії a_t у стані s_t;

γ – коефіцієнт дисконтування (між 0 та 1), який визначає важливість майбутніх винагород;

$Q(s_{t+1}, a_{t+1})$ – Q-значення для наступної пари стан-дія.

Пошук по дереву Монте-Карло – це евристичний метод прийняття рішень у задачах, що мають велику кількість можливих станів і дій, де побудова повного дерева рішень є обчислювально недоцільною. Метод поєднує методи імовірнісного моделювання (Монте-Карло) з пошуком у просторі рішень у вигляді дерева, дозволяючи ефективно досліджувати найбільш перспективні гілки дерева, з урахуванням балансу між дослідженням нових дій і використанням уже відомих хороших дій. На рисунку 2.15 показано схему поетапної роботи пошуку по дереву Монте-Карло.

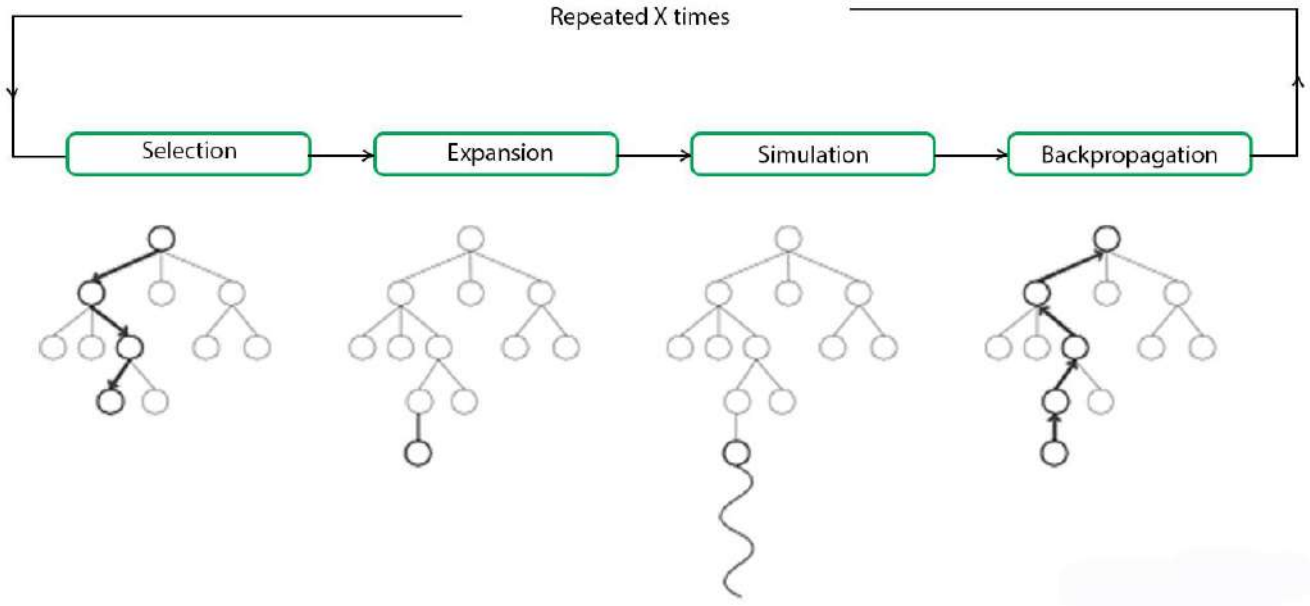


Рисунок 2.15 – Схема поетапної роботи пошуку по дереву Монте-Карло [75]

Алгоритм складається з чотирьох основних етапів: вибір, розширення, симуляція, зворотне оновлення. Етап вибору – починаючи з кореневого вузла, алгоритм проходить по дереву згідно з певною евристикою (зазвичай UCT), поки не досягне вузла, який ще не був повністю досліджений. На етапі розширення додається один або кілька нових дочірніх вузлів до поточного. На етапі симуляції виконується випадкове симульоване проходження з нового вузла до кінця гри або до певного глибини, щоб оцінити потенційну винагороду. І при зворотному оновленні результати симуляції поширюються назад угору по дереву, оновлюючи статистику відвідування та середні виграші для кожного вузла.

Також для вирішення питання планування та моніторингу маршрутів руху групи безпілотних літальних апаратів варто провести аналіз такої підгалузі машинного навчання як глибоке навчання.

Глибоке навчання (Deep Learning) базується на використанні глибоких нейронних мереж, здатних автоматично виявляти багаторівневі абстракції у великих обсягах даних [76]. Основна особливість глибокого навчання полягає в здатності моделі самостійно формувати представлення ознак, що дозволяє мінімізувати потребу у ручному доборі характеристик. Завдяки цьому глибоке навчання стало основою багатьох сучасних технологій від комп’ютерного зору й обробки природної мови до автономного керування складними технічними системами.

На відміну від класичних методів машинного навчання, глибокі нейронні мережі можуть мати десятки або навіть сотні прихованих шарів, кожен з яких виконує нелінійні перетворення вхідних даних. Це дає змогу моделі поступово виділяти все більш складні ознаки, виявляючи латентні залежності, які не піддаються лінійному аналізу.

До архітектур глибоких нейронних мереж, що можуть стати корисними для планування та моніторингу руху групи БПЛА належать: рекурентна нейронна мережа, згорткова нейронна мережа, трансформери, графові нейронні мережі.

Рекурентні нейронні мережі – це тип нейронних мереж, спеціально розроблений для обробки послідовних даних, де порядок і контекст відіграють

ключову роль [76]. На відміну від класичних нейронних мереж, де кожне входження обробляється незалежно, рекурентні нейронні мережі мають зворотні зв'язки, що дозволяють передавати інформацію про попередні стани в часовій послідовності. Це забезпечує здатність моделі враховувати контекст, накопичений під час обробки попередніх елементів послідовності, що критично важливо для аналізу тимчасових рядів, обробки природної мови, управління роботами, прогнозування динаміки тощо. На рисунку 2.16 показано базову архітектуру рекурентної нейронної мережі.

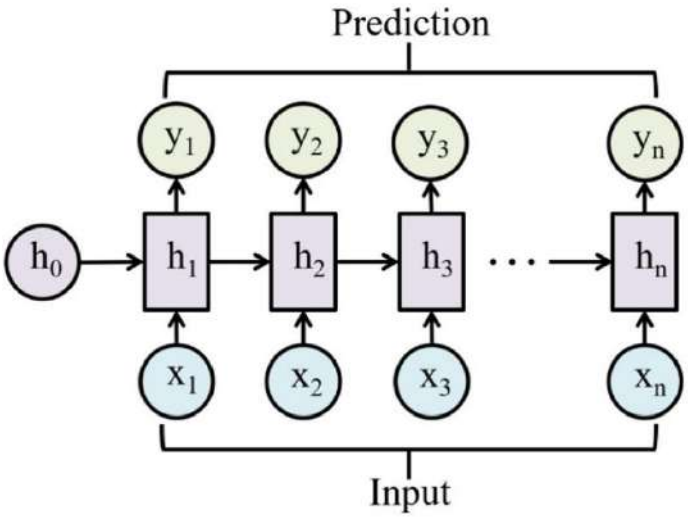


Рисунок 2.16 – Архітектура рекурентної нейронної мережі [77]

Проте стандартні рекурентні нейронні мережі мають обмежену здатність до запам'ятовування довготривалих залежностей, оскільки при тривалому зворотному поширенні градієнти мають схильність до затухання або вибуху. Для розв'язання цієї проблеми було запропоновано модифіковані архітектури, зокрема: LSTM та GRU.

LSTM (Long Short-Term Memory) – включає механізм керування пам'яттю за допомогою «воріт» (вхідного, вихідного та забуваючого), які контролюють, яка інформація повинна зберігатися, оновлюватися або видалятися. Завдяки цьому LSTM здатна ефективно моделювати довготривалі залежності та уникати проблеми зникнення градієнтів.

GRU (Gated Recurrent Unit) – спрощена версія LSTM із меншою кількістю параметрів, що зберігає більшість її переваг, водночас забезпечуючи швидше навчання і меншою обчислювальною складністю.

Згорткові нейронні мережі – це спеціалізований тип глибоких нейронних мереж, який демонструє високу ефективність у задачах обробки просторово організованих даних, зокрема зображень, відео, а також тривимірних структур [76]. Основною особливістю згорткових нейронних мереж є застосування згорткових шарів, які автоматично виявляють локальні ознаки, такі як краї, кути, текстури або інші просторові шаблони. Це дозволяє моделі ефективно вчитися на зображеннях з великою кількістю пікселів, значно зменшуючи кількість параметрів порівняно з повнозв’язними мережами.

Архітектура згорткових нейронних мереж зазвичай включає чергування згорткових шарів, функцій активації (наприклад, ReLU), шарів субдискретизації (pooling) та повнозв’язаних (dense) шарів. Згорткові шари витягують ознаки з вхідних зображень, тоді як pooling-шари зменшують просторову розмірність ознак, знижуючи обчислювальні витрати та підвищуючи стійкість до зсувів і шуму. На рисунку 2.17 показано просту архітектуру згорткової нейронної мережі.

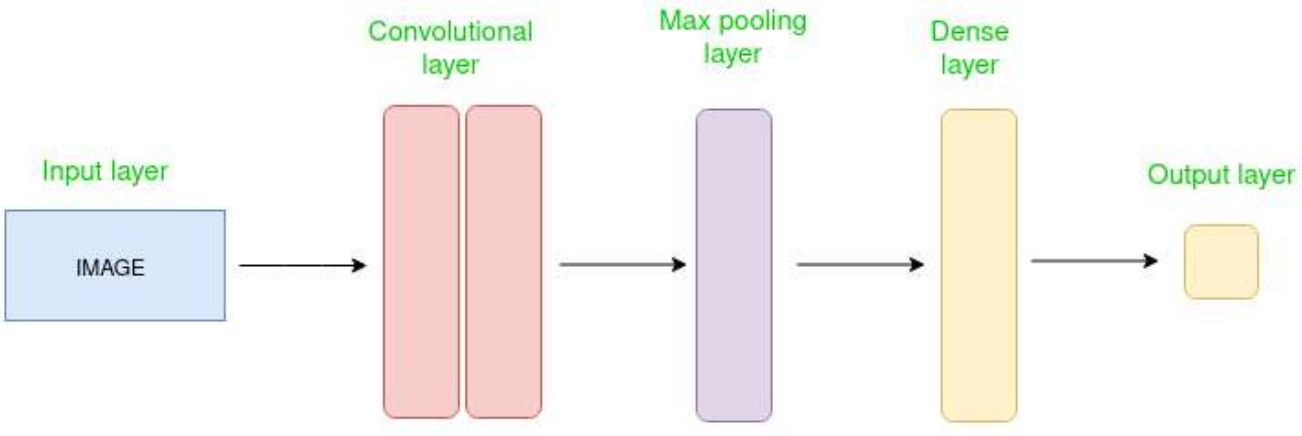


Рисунок 2.17 – Архітектура згорткової нейронної мережі [78]

Трансформери – це архітектура глибоких нейронних мереж, яка набула широкого застосування у задачах моделювання послідовностей та прийняття

рішень на основі контексту. Вперше запропонована для обробки природної мови, архітектура трансформерів продемонструвала високу ефективність у різноманітних галузях, зокрема в управлінні автономними агентами, робототехніці та системах прогнозування. Її ключовими перевагами є здатність моделювати складні залежності між елементами вхідної послідовності, паралельна обробка даних та хороша масштабованість. Однією з головних особливостей трансформерів є використання позиційного кодування, що забезпечує збереження інформації про порядок елементів у послідовності. Це дозволяє моделі враховувати часову динаміку станів.

Графові нейронні мережі – це клас нейронних мереж, які працюють з даними, представленими у вигляді графів [79]. Вони особливо корисні, коли об'єкти та їх взаємозв’язки можна описати у вигляді вузлів і ребер графа. Кожен вузол оновлює свій стан, агрегуючи інформацію від своїх сусідів у графі. Оновлені вузлові представлення пропускаються через нейронні шари (зазвичай вони повнозв’язні). Процес повторюється кілька разів, щоб інформація поширювалась по всьому графу. Графові нейронні мережі можуть стати корисними для : моделювання взаємодії між БПЛА, координації руху, аналізу мережі зв’язку, планування маршрутів.

З розвитком обчислювальної техніки особливої популярності набули методи глибокого навчання з підкріпленням (Deep Reinforcement Learning, DRL), які поєднують класичні підходи з глибокими нейронними мережами. До таких належать: глибока Q-мережа (DQN), глибокий детерміністський градієнт політики (DDPG), проксимальна оптимізація політики (PPO), A3C.

Глибока Q-мережа (DQN) – поєднує Q-Learning із глибинними нейронними мережами для апроксимації Q-функції у великих просторах станів [80]. У традиційному Q-Learning зберігається таблиця значень Q для кожної пари стан–дія, що є ефективним лише для малих або дискретних просторових моделей. Проте у складних або неперервних середовищах, зокрема у випадках управління БПЛА, кількість можливих станів та дій настільки велика, що табличне подання стає непрактичним. Глибока Q-мережа вирішує цю проблему, замінюючи таблицю Q-

функції на нейронну мережу, яка навчається прогнозувати значення Q для заданої пари стану і дії.

Архітектура DQN складається з глибинної нейронної мережі, що приймає на вхід вектор ознак поточного стану та видає оцінку очікуваної вигоди (Q-значення) для кожної можливої дії. Під час навчання модель оновлює свої ваги за допомогою градієнтного спуску на основі різниці між прогнозованим та цільовим Q-значенням, що визначається за допомогою формули Беллмана. Для стабілізації процесу навчання в DQN застосовуються додаткові техніки, такі як Replay Memory (буфер досвіду) для випадкового вибору досвіду з минулих ітерацій, та Target Network – окрема мережа, яка оновлюється повільніше і використовується для генерації цільових значень. На рисунку 2.18 показано структуру глибокої Q-мережі.

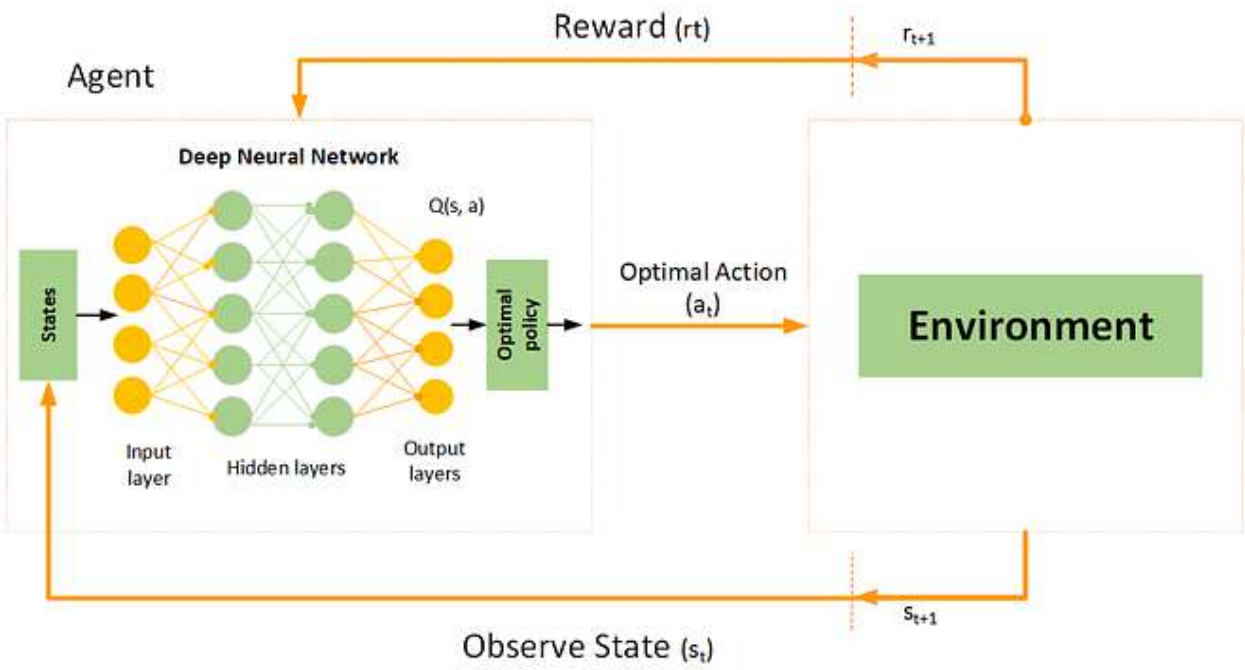


Рисунок 2.18 – Структура глибокої Q-мережі [81]

Глибокий детерміністський градієнт політики (DDPG) – метод для задач з неперервними просторами дій [82]. Він поєднує глибокі нейронні мережі з актор-критик підходом, де актор формує політику дій, а критик оцінює їхню якість. На відміну від дискретних алгоритмів, таких як DQN, DDPG дозволяє безпілотним

літальним апаратам обирати дії з неперервного діапазону, наприклад, змінювати швидкість, напрямок або висоту з високою точністю. Для стабільного навчання застосовуються буфер досвіду (replay buffer) та окремі цільові мережі. Завдяки цьому DDPG є ефективним для керування рухом БПЛА в складних динамічних середовищах, де необхідна точна та плавна координація дій.

Проксимальна оптимізація політики (PPO) – алгоритм який забезпечує стабільне та ефективне оновлення політики [83]. Його головна особливість це обмеження величини змін у політиці між ітераціями навчання, що запобігає різким коливанням і покращує збіжність. PPO використовує обмежену функцію втрат, яка балансує між дослідженням та експлуатацією, зберігаючи стабільність процесу навчання. Завдяки простоті реалізації та високій надійності, алгоритм широко застосовується у задачах керування роботами, включно з управлінням рухом БПЛА, де важливо швидко адаптуватися до змін у середовищі та координувати дії в реальному часі.

A3C (Asynchronous Advantage Actor-Critic) – це алгоритм який реалізує акторно-критичну архітектуру з використанням кількох паралельних агентів [84]. Кожен агент навчається незалежно у власному середовищі, асинхронно оновлюючи спільні параметри нейронної мережі. Такий підхід значно прискорює процес навчання та зменшує кореляцію між зібраними даними. A3C добре підходить для задач із великими та складними просторами станів, зокрема для управління групою БПЛА, де потрібна швидка адаптація до динамічного середовища та ефективна координація дій між агентами.

У задачах управління кількома агентами (наприклад, групою БПЛА) також застосовується мультиагентне навчання з підкріпленням (MARL). Відповідно, виникають архітектурні підходи, які забезпечують узгодженість дій: CTDE, MADDPG, COMA.

CTDE (Centralized Training with Decentralized Execution) – централізоване навчання із децентралізованим виконанням, що дозволяє ефективно координувати агентів у складному середовищі;

MADDPG (Multi-Agent Deep Deterministic Policy Gradient) – розширення DDPG для мультиагентного сценарію;

COMA (Counterfactual Multi-Agent Policy Gradients) – підхід, що враховує внесок кожного агента в загальну винагороду.

2.2 Порівняння приведених алгоритмів

Щоб вирішити поставлену задачу, потрібно обрати лише один алгоритм, тому проаналізуємо переваги й недоліки основних підходів машинного навчання, оберемо той, що найкраще задовольняє наші потреби і тоді проаналізуємо кожен алгоритм з нього та розглянемо їхню практичну придатність.

Контрольоване навчання передбачає навчання моделі на мічених даних, де відома відповідь (мітка) для кожного прикладу. Цей підхід є ефективним для класифікації та регресії, але малопридатний у випадках, коли дані є динамічними, а навчальні мітки відсутні або не можуть бути точно визначені. Для задачі автономного планування руху групи БПЛА, де середовище змінюється, а результати дій відомі лише після їх виконання, цей підхід виявляється непридатним.

Неконтрольоване навчання використовується для виявлення прихованих структур у даних без наявності міток. Незважаючи на корисність у кластеризації або зниженні розмірності, цей підхід не забезпечує можливості навчання послідовній взаємодії агента з середовищем у реальному часі.

Машинне навчання з підкріпленням, натомість, орієнтоване саме на взаємодію агента з динамічним середовищем шляхом спроб і помилок, що супроводжується винагородою або штрафом. Цей підхід дозволяє агенту самостійно навчатися найкращій стратегії поведінки, з урахуванням довготривалих наслідків своїх дій. Саме через свою здатність до адаптивного прийняття рішень у реальному часі, роботи в умовах часткової інформації та відсутність необхідності мічених даних, підхід з підкріпленням є найдоцільнішим вибором для реалізації системи планування руху БПЛА.

Серед розглянутих алгоритмів, кожен має свої переваги та обмеження в контексті керування групою БПЛА.

Глибока Q-мережа (DQN) демонструє високу ефективність у дискретних середовищах, однак непридатний для неперервного простору дій, що обмежує його застосування у задачах навігації БПЛА.

Глибокий детерміністський градієнт політики (DDPG) підтримує неперервні дії та добре масштабується, але є чутливим до вибору гіперпараметрів, що часто призводить до нестабільного навчання.

Проксимальна оптимізація політики (PPO) забезпечує високу стабільність, стійкість до перенавчання та добре працює у складних мультиагентних середовищах.

A3C дозволяє асинхронне навчання та пришвидшену збіжність, проте поступається PPO за стабільністю та має вищі вимоги до ресурсів під час реалізації.

При виборі архітектур навчання для мультиагентної системи БПЛА слід ураховувати особливості інформаційного доступу, координації та обчислювальних обмежень.

CTDE є універсальною основою, що дозволяє навчатись із повною інформацією та виконуватись локально. Цей підхід балансує між продуктивністю та автономністю, забезпечуючи стійке масштабування системи.

MADDPG добре працює у неперервних просторах дій і є ефективним для невеликої кількості агентів, проте гірше масштабується при зростанні числа учасників.

COMA вирішує проблему розподілу командної винагороди, але має високу обчислювальну складність, що ускладнює використання в системах з великою кількістю БПЛА.

З урахуванням вищезазначеного, найбільш доцільною комбінацією для реалізації системи планування та моніторингу руху групи БПЛА є алгоритм PPO у зв’язці з архітектурою CTDE. Такий підхід поєднує високу стабільність навчання, адаптивність до змін середовища та можливість децентралізованого виконання, що критично важливо для автономної роботи мультиагентної системи.

2.3 Висновки до другого розділу

У цьому розділі було розглянуто теоретичні основи машинного навчання, його основні типи та сучасні підходи, які становлять базу для побудови інтелектуальних систем автономного керування групами БПЛА. Детально охарактеризовано контрольоване, неконтрольоване та навчання з підкріпленням, розкрито сутність глибокого навчання й описано найпоширеніші архітектури глибоких нейронних мереж, зокрема згорткові та рекурентні моделі.

Також було визначено практичну цінність застосування алгоритмів машинного навчання в задачах управління БПЛА, зокрема для планування маршруту, ухвалення рішень у реальному часі, обробки сенсорних даних та забезпечення узгодженої роботи групи апаратів.

Окрім цього, виконано порівняльний аналіз основних підходів машинного навчання з метою обґрунтування вибору найпридатнішого для задачі автономного планування руху БПЛА. У результаті обґрунтовано доцільність використання навчання з підкріпленням, а також проведено порівняння низки актуальних алгоритмів цього підходу та відповідних мультиагентних архітектур. Встановлено, що оптимальним вибором для реалізації запропонованої системи є поєднання алгоритму проксимальної оптимізації політики (PPO) з архітектурою CTDE, що забезпечує баланс між стабільністю, масштабованістю та ефективністю децентралізованого керування.

Отримані теоретичні знання створюють основу для реалізації запропонованої системи планування та моніторингу руху групи БПЛА.

3 МОДЕЛІ ТА МЕТОДИ ПЛАНУВАННЯ ТРАЄКТОРІЙ РУХУ ГРУПИ БПЛА В УМОВАХ НЕПОВНОЇ ІНФОРМАЦІЇ

3.1 Багатоагентне планування маршрутів: постановка задачі та виклики

Планування маршруту для декількох БПЛА можна розглядати як задачу багатоагентного планування маршруту (Multi-Agent Path Finding, MAPF) [85], яка є моделлю, що використовується для знаходження оптимального шляху для декількох агентів від стартових позицій до пунктів призначення без конфліктів. По суті, MAPF є відносно складною задачею оптимізації зі спільною метою. Простір станів цієї задачі експоненціально зростає з кількістю агентів, і було доведено, що вона є NP-важкою задачею [86]. У завданнях розвідки групи БПЛА повинні не тільки уникати небезпечних зон і безпечно досягати цільових точок, але й покривати більшу територію за коротший час. Однак витрати часу і площа покриття суперечать один одному, оскільки це багатоцільові оптимізаційні задачі, і ми повинні знайти компроміс між двома або більше суперечливими цілями, щоб прийняти оптимальне рішення. Неможливо знайти рішення, яке дозволить досягти оптимального виконання всіх цілей, тому для багатоцільової оптимізаційної задачі зазвичай використовують множину непогіршених розв'язків, яку називають "множиною розв'язків Парето" [87].

3.2 Використання навчання з підкріпленням у плануванні маршрутів БПЛА

В останні роки, зі стрімким розвитком глибокого навчання з підкріпленням (Deep Reinforcement Learning, DRL), його потужні можливості представлення і навчання дозволили йому добре працювати в задачах прийняття рішень [88], тому дослідники починають вивчати застосування навчання з підкріпленням в плануванні траєкторії руху і навігації групи БПЛА [89]. Порівняно з традиційними алгоритмами, навчання з підкріпленням працює краще, коли середовище невідоме

і динамічне. Крім того, швидкість висновку та узагальнення навчання з підкріпленням є перевагами в задачах прийняття рішень в реальному часі.

У цьому дослідженні можливості сприйняття групи БПЛА були обмежені, а спостереження за навколишнім середовищем проводилися лише частково, що означає, що групи БПЛА було важко приймати оптимальні рішення, коли глобальні стани були відсутні, оскільки переходи між станами були невідомими. Дії кожного БПЛА можуть змінити стан середовища, особливо в алгоритмі, що базується на навчанні, наприклад, на навчанні з підкріпленням, неповна інформація призведе до низької ефективності та збіжності. Більше того, життєво важливо розробити навчальну архітектуру для координації декількох БПЛА для досягнення спільної мети. Нерозумно застосовувати повністю розподілену архітектуру навчання для вирішення проблем MAPF через високу складність. Те ж саме стосується і багатоцільових задач оптимізації. Було запропоновано структуру централізованого навчання з децентралізованим виконанням [90], що дозволяє використовувати додаткову інформацію в політиках для полегшення навчання. Було доведено, що цей фреймворк здатен розв'язувати спільні задачі, такі як багатоагентне планування шляхів.

Крім того, функція винагороди навчання з підкріпленням повинна бути переглянута в світлі багатоцільових проблем оптимізації. З одного боку, занадто проста функція винагороди може спричинити "злам винагороди" [91] і посилити труднощі навчання політики через неповну інформацію. З іншого боку, занадто складна функція винагороди призведе до гіршого узагальнення. Найпоширенішим рішенням є розробка винагороди, що задовольняє декілька цілей з різною вагою відповідно до попередніх знань, в якому багатоцільова задача оптимізації буде змінена на одно-цільову задачу оптимізації. Фактично, цей розв'язок є близьким до оптимального. При плануванні маршрутів для декількох БПЛА необхідно враховувати деякі обмеження, такі як витрати часу, безпека і покриття.

Для вирішення цих проблем було запропоновано вдосконалений мультиагентний алгоритм навчання з підкріпленням, заснований на централізованому навчанні та децентралізованій архітектурі виконання. Алгоритм

навчання політики – це проксимальна оптимізація політики (PPO, Proximal Policy Optimization). Це безмодельний алгоритм навчання з підкріпленням, який може адаптуватися до динамічного середовища і забезпечує хороше узагальнення. Мережа критиків (Critic Network) PPO використовується для координації всіх БПЛА, щоб максимізувати віддачу команди за допомогою централізованого навчання шляхом отримання спільних спостережень, а мережа акторів (Actor Network) PPO використовується для виведення дій. Було додано рекурентну нейронну мережу до акторно-критичної мережі для збору історичної інформації з прихованого стану рекурентної нейронної мережі, що вирішує проблему неповної інформації, спричиненої частковими спостереженнями. Крім того, розроблено спільну функцію винагороди, щоб допомогти групам БПЛА навчитися оптимальній політиці. Після завершення етапу навчання кожен БПЛА може виконати дію на основі своїх локальних спостережень на еталонному етапі.

Коли декілька БПЛА виконують розвідувальні місії, вони повинні приймати рішення в реальному часі на основі поточної інформації про стан, а також планувати шлях до цільових точок без зіткнень. Крім того, необхідно враховувати часові витрати і покриття. Таким чином, автономне планування маршруту для декількох БПЛА є проблемою прийняття рішень в режимі онлайн в умовах неповної інформації. Вона має три характеристики: розподілене прийняття рішень, часткове спостереження і багатоцільова оптимізація. Автономне планування маршруту для декількох БПЛА вважається повністю кооперативною задачею. Метою всіх учасників такої задачі є отримання максимального командного прибутку. Тому ми можемо створити мультиагентну модель послідовного прийняття рішень в реальному часі за допомогою теоретичної бази децентралізованого частково спостережуваного марковського процесу прийняття рішень (Dec-POMDP, Decentralized Partially Observable Markov Decision Process) [92].

Модель Dec-POMDP – це загальна модель, яка використовується для розв'язання багатоагентних задач оптимізації цілей у кооперативних середовищах, яка узагальнює частково спостережуваний марковський процес прийняття рішень

(POMDP) для багатоагентних середовищ. Він дозволяє здійснювати розподілений контроль над декількома агентами, які можуть не мати можливості спостерігати глобальний стан середовища. На кожному кроці кожен агент обирає дію на основі локальних спостережень (всі агенти паралельно), а потім отримує власну винагороду від середовища, і всі агенти співпрацюють для отримання спільних довгострокових вигод і максимізації прибутку. Як правило, модель Dec-POMDP описується кортежем:

$$\left\langle I, S, \{A_i\}, T, R, \{\Omega_i\}, O, h \right\rangle, \tag{3.1}$$

де I – це набір з N агентів;

S – множина станів середовища, а S_0 - початковий стан;

$\{A\}$ – множина дій агентів. Це кортеж дій A_1, A_2,...,A_i;

T – функція ймовірності переходу стану $P(S' \mid S, A)$;

R – винагорода, коли агенти виконують дії $\{A\}$ у стані S, вона залежить від усіх агентів;

$\{\Omega_i\}$ – набір спостережень для агентів;

O – таблиця ймовірностей спостереження, де $O(o_1, o_2, ..., o_i \mid S', A)$ – ймовірність того, що $O(o_1, o_2, ..., o_i \mid S', A)$ спостерігаються всіма агентами відповідно;

h – максимальна кількість кроків в епізоді, який називається «горизонт».

Однак складність оптимального розв'язку цієї розподіленої моделі є подвійною експоненціальною (формула 3.2) [93] тому її важко обчислити безпосередньо, і для отримання наближеного розв'язку зазвичай використовують навчання з підкріпленням. Зокрема, вона оцінюється як:

$$O\left[\left(\left|\mathrm{A}\right|^{\frac{|o|^h - 1}{|o| - 1}}\right)^n\right] \tag{3.2}$$

3.3 Алгоритм актор-критик

При навчанні з підкріпленням агент безперервно взаємодіє з навколишнім середовищем, щоб оптимізувати політику за допомогою зворотного зв'язку (винагороди), що надається середовищем. Навчання з підкріпленням в основному поділяється на методи, засновані на цінностях, і методи, засновані на політиці. Алгоритм на основі градієнта політики може легко вибрати відповідну дію в безперервному просторі дій, тоді як алгоритм на основі цінностей не може. Однак обмеженням алгоритму градієнта політики є його низька ефективність навчання. Тому дослідники запропонували метод, який поєднує алгоритм градієнта політики та алгоритм, заснований на цінностях, який отримав назву алгоритм акторної критики [94]. Архітектура actor-critic показана на рисунку 3.1. Актор-критик використовує мережу, засновану на цінностях, і мережу, засновану на політиці, як мережу критика і мережу актора, відповідно. Мережа критиків може реалізувати однокрокові оновлення для подолання низької ефективності навчання, а мережа акторів виводить дії відповідно до поточного спостереження, в той час як мережа критиків може судити, чи є поточна дія хорошою чи поганою, що може призвести до того, що мережа акторів виводить кращу дію. В даний час алгоритми, засновані на акторно-критичній структурі, такі як Deep Deterministic Policy Gradient (DDPG), Proximal Policy Optimization (PPO) і Asynchronous Advantage Actor-Critic (A3C), є дуже популярними.

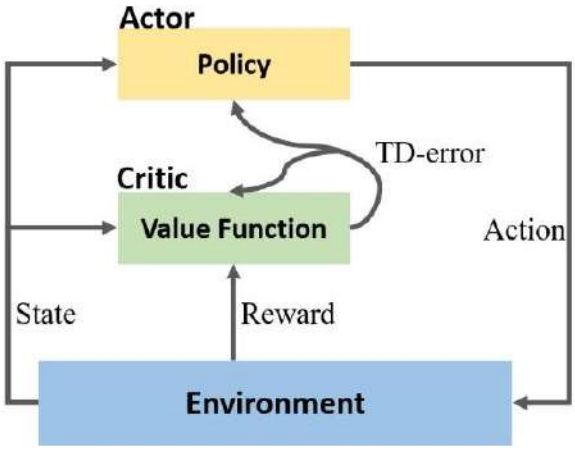


Рисунок 3.1 – Архітектура алгоритму актор-критик [94]

3.4 Централізоване навчання та децентралізована архітектура виконання

Коли є кілька агентів у повністю кооперативному середовищі, ми можемо створити фреймворк Dec-POMDP. Навчання з підкріпленням – чудовий метод пошуку оптимального рішення моделі. Однак, якщо ми безпосередньо використовуємо одноагентні алгоритми навчання з підкріпленням для навчання агентів незалежно один від одного, їх важко зблизити, оскільки дії кожного агента будуть змінювати середовище, тобто середовище буде нестабільним для кожного агента і призведе до труднощів у навчанні. Алгоритм Multi-Agent Deep Deterministic Policy Gradient (MADDPG) навчає мультиагентів за допомогою централізованої мережі критиків, що є гарним рішенням для навчання мультиагентних систем. Як показано на рисунку 3.2, на вхід мережі критиків подаються спільні спостереження всіх агентів у середовищі на етапі навчання, а мережа акторів вводить лише власні локальні спостереження і виводить дії відповідно до цих спостережень на етапі виведення. Така архітектура дозволяє кожному агенту спостерігати за діями інших агентів у середовищі, забезпечуючи стабільність середовища. Тому мультиагентне навчання з підкріпленням здебільшого базується на архітектурі з централізованим навчанням і децентралізованим виконанням (CTDE).

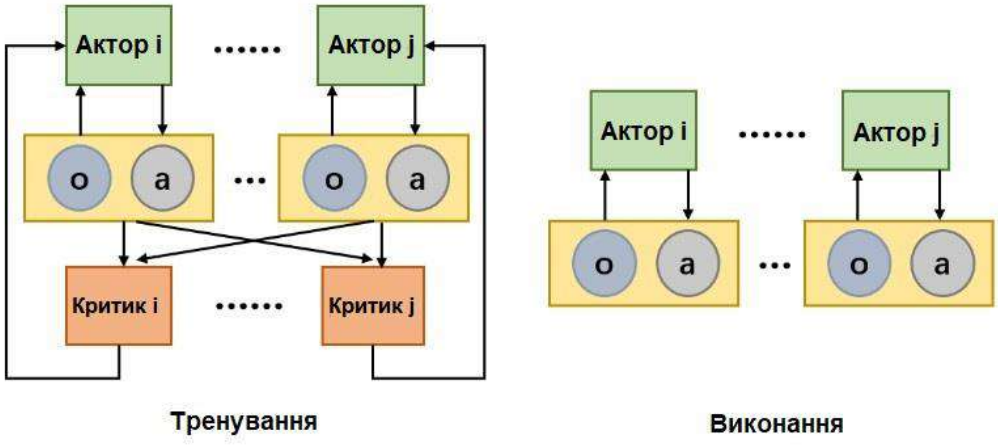


Рисунок 3.2 – Централізована підготовка (ліворуч) та децентралізоване виконання (праворуч)

3.5 Проксимальна оптимізація політики за допомогою CTDE

У даному дослідженні для керування БПЛА в рамках політики навчання обрано алгоритм проксимальної оптимізації політики (PPO). PPO є алгоритмом навчання з підкріпленням на основі політики, який поєднує ефективність градієнта політики з стабільністю оптимізації, притаманною методам з обмеженням оновлення політики. Цей алгоритм базується на акторно-критичній архітектурі (actor-critic), яка добре масштабується для задач з високовимірним безперервним простором дій, що робить його особливо придатним для контролю траєкторій БПЛА в реальному часі.

Алгоритми на основі градієнта політики зазвичай мають високу варіативність і нестабільність у процесі навчання. Причина полягає в тому, що при надмірному оновленні політики між ітераціями (тобто при великій різниці між старою та новою політикою) навчання стає нестабільним, що може призвести до втрати вже здобутої інформації або зниження продуктивності агента.

Щоб уникнути цих проблем, було запропоновано нову цільову функцію з обмеженням оновлення політики, яку можна оновлювати невеликими партіями за кілька кроків навчання, що вирішує проблему складності визначення кроків в алгоритмі градієнта політики. Алгоритм враховує різницю між старою та новою мережею при оновленні параметрів. Основою PPO є градієнт очікуваної винагороди, який визначається як:

$$\nabla \overline{R}(\tau) = \mathrm{E}_{\tau \sim \pi\theta(\tau)}\left[A^{\pi}(s_t, a_t) \nabla \log p_{\theta}(a_t \mid s_t) \right], \tag{3.3}$$

де π_θ – це політика, параметризована вектором θ;

$A^{\pi}(s_t, a_t)$ – функція переваги.

Функція переваги визначає наскільки конкретна дія краща за середню в даному стані і вона обчислюється як:

$$A^{\pi}(s,a) = Q^{\pi}(s,a) - V^{\pi}(s), \tag{3.4}$$

де $Q^{\pi}(s,a)$ – очікувана сумарна винагорода після виконання дії a а у стані s;

$V^{\pi}(s)$ – очікувана винагорода зі стану s за політикою π.

Щоб забезпечити стабільність оновлень, PPO використовує техніку переваги важливості (importance sampling):

$$\mathbb{E}_{x\sim p}[f(x)] = \mathbb{E}_{x\sim q}\left[f(x)\frac{p(x)}{q(x)}\right], \tag{3.5}$$

яка дозволяє переоцінювати очікування за новою політикою, використовуючи зібрані дані зі старої.

Головною особливістю PPO є використання "обмеження" (clipping) для стабілізації навчання. Це дозволяє запобігти надмірним змінам у політиці. Відповідна обмежена цільова функція має вигляд:

$$\mathcal{L}^{CLIP}(\theta) = \mathbb{E}_t\left[\min\left(\left(\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{old}}(a_t \mid s_t)}\right)\hat{A}_t, \text{clip}\left(\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{old}}(a_t \mid s_t)}, 1-\epsilon, 1+\epsilon\right)\hat{A}_t\right)\right], \tag{3.6}$$

де $\hat{A}_t$ – це оцінка функції переваги на кроці t;

ϵ – визначає допустиму межу зміни політики.

Такий підхід забезпечує збалансованість між дослідженням нових дій і збереженням ефективної стратегії, що особливо важливо для контролю багатьох БПЛА в реальному середовищі.

На основі алгоритму PPO було застосовано метод навчання архітектури CTDE (Centralized Training with Decentralized Execution) і розроблено

багатоагентний алгоритм PPO в багатоагентному середовищі. У порівнянні з одноагентним середовищем, вхідною інформацією для мережі-критика є спільне спостереження декількох БПЛА, що еквівалентно роботі централізованого контролера. Такий підхід дозволяє кожному дрону отримувати більше релевантної інформації для прийняття рішень. Оновлення мережі актора здійснюється шляхом максимізації наступної цільової функції:

$$\mathcal{L}(\theta) = \frac{1}{Bn}\sum_{i=1}^{B}\sum_{k=1}^{n}\left[\min\left(r_{\theta,i}^{k}A_{i}^{k}, \text{clip}\left(r_{\theta,i}^{k}, 1-\epsilon, 1+\epsilon\right)A_{i}^{k}\right) + \sigma \cdot S_{\pi}\right], \tag{3.7}$$

де $r_{\theta,i}^{k} = \dfrac{\pi_{\theta}(a_i^k \mid o_i^k)}{\pi_{\theta_{\text{old}}}(a_i^k \mid o_i^k)}$ – це відношення ймовірностей між новою та старою політикою;

A_i^k – функція переваги;

ϵ – параметр обмеження;

S_{π} – ентропія політики, що сприяє дослідженню нових дій.

Мережа критика, у свою чергу, оновлюється шляхом мінімізації втрат функції цінності, яка задається наступним чином:

$$\mathcal{L}(\phi) = \frac{1}{Bn}\sum_{i=1}^{B}\sum_{k=1}^{n}\max\left[\left(\mathcal{V}_{\phi}(s_i^k) - \mathcal{R}_i\right)^2, \left(\text{clip}\left(\mathcal{V}_{\phi}(s_i^k), \mathcal{V}_{\phi_{\text{old}}}(s_i^k) - \varepsilon, \mathcal{V}_{\phi_{\text{old}}}(s_i^k) + \varepsilon\right) - \mathcal{R}_i\right)^2\right], \tag{3.8}$$

де $\mathcal{R}_i^k$ – це цільове значення (наприклад, виграш або оцінка повернення);

$\mathcal{V}_{\phi}(s)$ – значення стану, оцінене поточною мережею-критиком.

Ваги двох нейронних мереж (акторської та критичної) оновлюються в кожному епізоді. Процес оновлення представлено на рисунку 3.3.

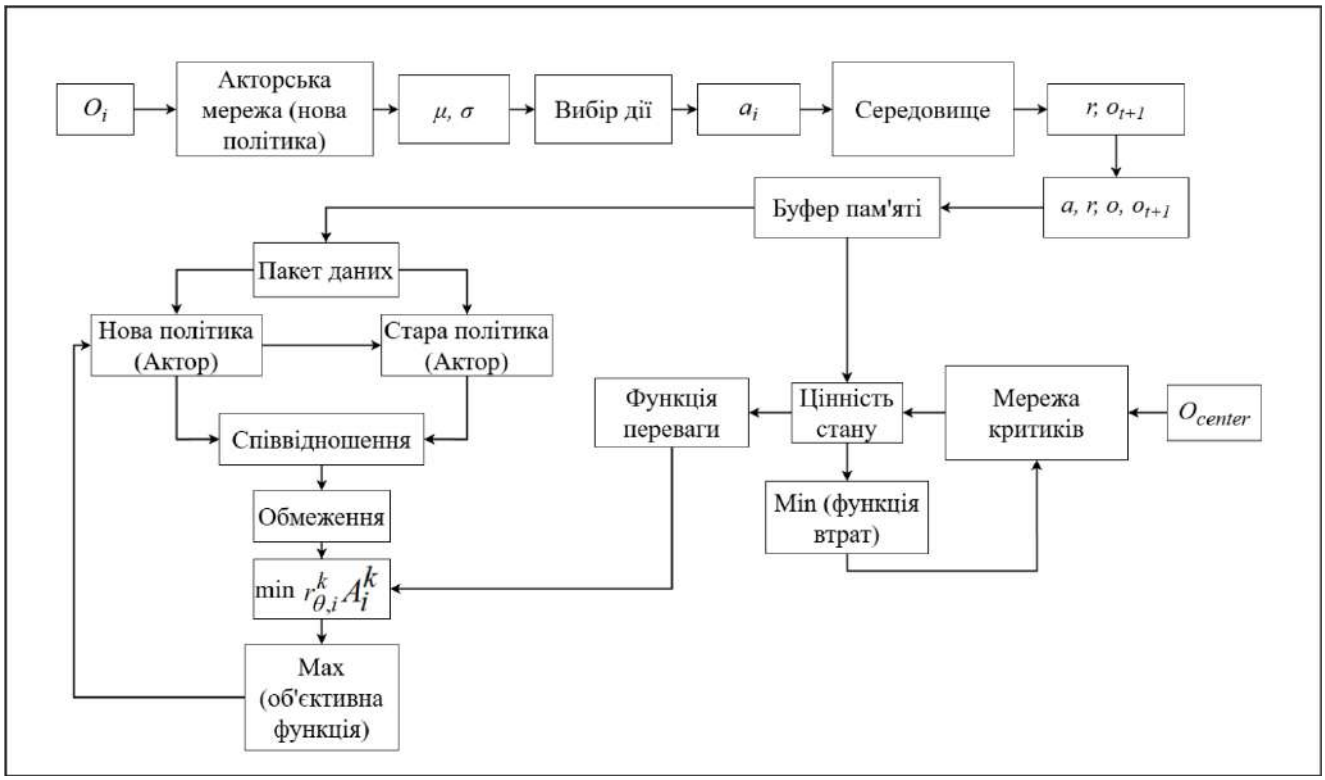


Рисунок 3.3 – Схема оновлення ваг мережі актор-критик у кожному епізоді

Акторська мережа, отримуючи на вхід стан середовища o_t, генерує параметри нормального розподілу μ, σ, з яких вибирається дія a_t. Після виконання дії середовище повертає винагороду r_t і новий стан o_{t+1}. Зібрані дані зберігаються в буфер пам’яті, після чого відбувається формування пакету даних (batch) та обчислення переваги (advantage function). Паралельно оновлюється критична мережа для мінімізації втрат. Далі обчислюється співвідношення між старою та новою політиками, застосовується операція обмеження змін в політиці (clipping), і вибирається максимум з об'єктивної функції. Це дозволяє стабілізувати навчання та уникнути великих оновлень політики.

3.6 Додавання шару RNN для неповної інформації

Однією з труднощів в задачах автономного планування маршруту з декількома БПЛА є часткове спостереження, що призводить до обмеженої інформації, яку отримують БПЛА. Вирішенням цієї проблеми є використання попереднього стану, щоб уникнути потрапляння в локальний оптимум.

Рекурентні нейронні мережі (RNN) можуть запам'ятовувати попередню інформацію і застосовувати її обчислення поточного виходу. Вузли між прихованими шарами з'єднані, а вхід прихованого шару включає поточний вхід і попередній вихід а також вихід прихованого шару в даний момент.

Дослідження глибокого рекурентного Q-навчання (Deep Recurrent Q-Network, DRQN) було першим, яке поєднало рекурентні мережі з методами навчання з підкріпленням. Як показано на рисунку 3.4, DRQN по суті перетворює один з лінійних шарів DQN на шар RNN (наприклад, LSTM або GRU). Це дозволило моделі зберігати інформацію про попередні стани середовища і досягати високих результатів навіть без стекування кадрів, як це було потрібно в класичному DQN в іграх від «Atari Games».

Після натхнення успіхами DRQN, у даному дослідженні було додано шар RNN до нейронної мережі в алгоритмі PPO. Це дозволило використати його для обробки послідовності історичних спостережень, щоб вирішити проблему неповної інформації в процесі навчання.

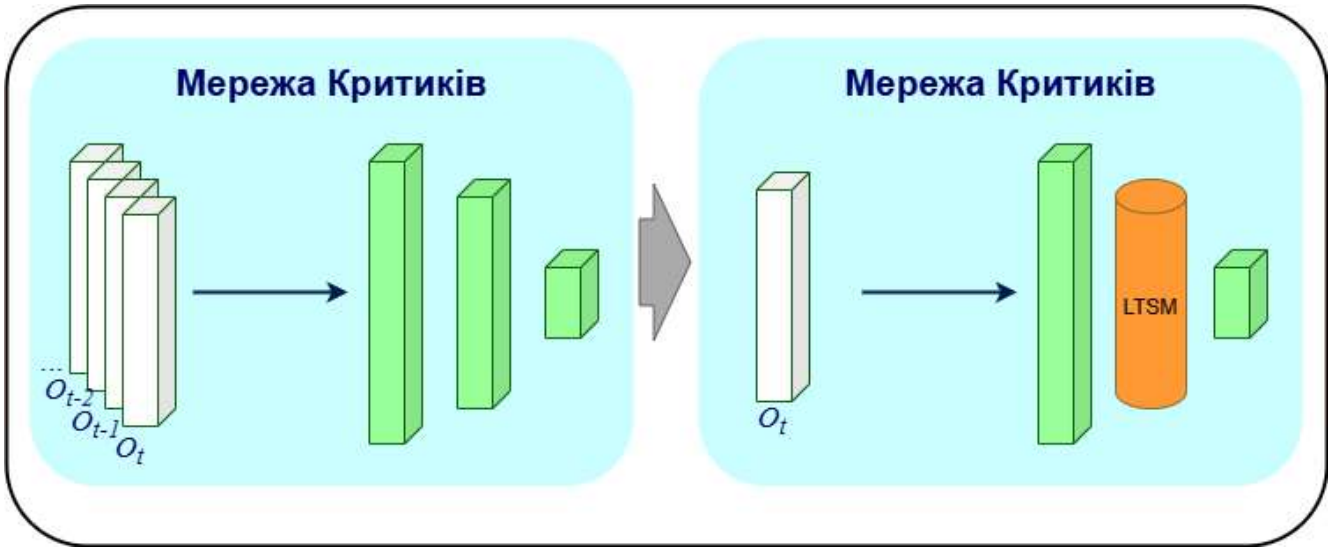


Рисунок 3.4 – Додавання рекурентного шару LSTM у архітектуру мережі критика

Хоча рекурентні нейронні мережі (RNN) є ефективним інструментом для роботи з послідовними даними, вони мають суттєве обмеження – короткотермінову пам’ять. Якщо послідовність є занадто довгою, інформація з ранніх кроків

практично не впливає на результати на пізніших етапах. Причиною цього є затухання градієнта під час зворотного поширення помилки (backpropagation through time, BPTT), що ускладнює ефективне навчання моделі на довгих залежностях у часі.

Для вирішення цієї проблеми було запропоновано архітектуру Long Short-Term Memory (LSTM) – модифікацію класичного RNN, яка дозволяє зберігати важливу інформацію довше завдяки використанню механізму "воріт" (gates). LSTM має спеціальну структуру, яка дозволяє моделі самостійно вирішувати:

– яку інформацію потрібно забути (forget gate);
– яку нову інформацію запам’ятати (input gate);
– яку інформацію вивести на наступний крок (output gate).

Ці ворота реалізуються за допомогою сигмоїдних і тангенсоїдних активацій, які дозволяють динамічно регулювати потік інформації. Завдяки цьому LSTM здатен утримувати важливі залежності в довгих послідовностях, запобігати затуханню градієнтів, і, як наслідок, покращувати ефективність навчання в умовах складної часової динаміки. На рисунку 3.5 зображено структуру блоку LSTM

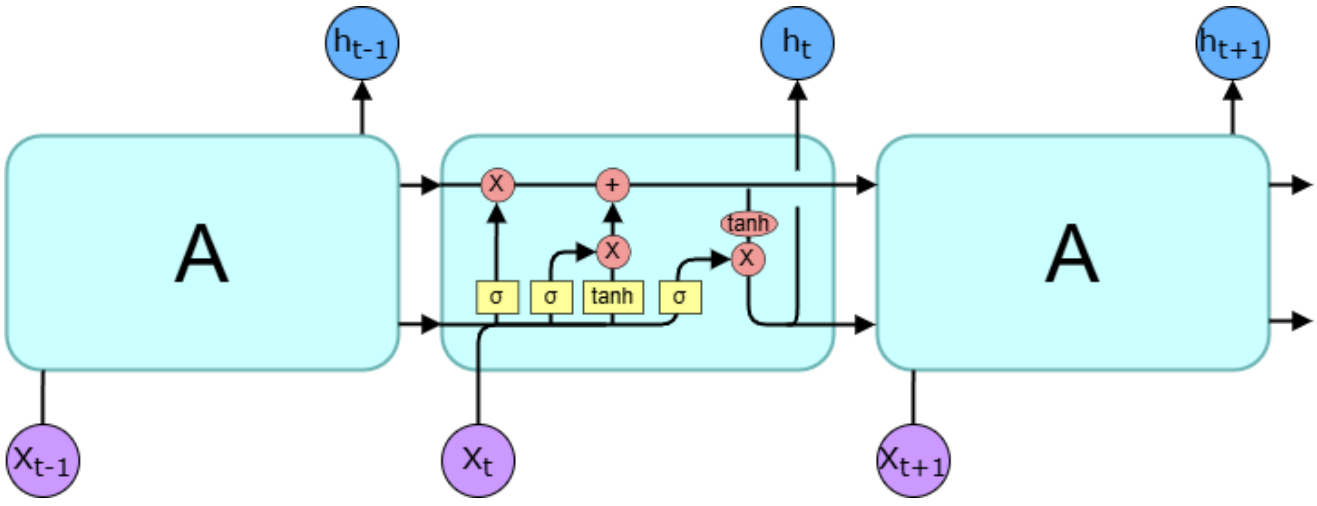


Рисунок 3.5 – Структура LSTM: у момент часу t, X_t – вхідний сигнал, C_t – стан комірки, h_t – прихований стан

На кожному кроці часу t, вхід x_t обробляється спільно з прихованим станом h_{t-1} та комірковим станом C_{t-1}, що дозволяє генерувати оновлені значення h_t та C_t:

$$\begin{cases} f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \\ i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right) \\ \tilde{C}_t = \tanh\left(W_c \cdot [h_{t-1}, x_t] + b_c\right) \\ C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \\ o_t = \sigma\left(W_o \cdot [h_{t-1}, x_t] + b_o\right) \\ h_t = o_t * \tanh(C_t) \end{cases}, \tag{3.9}$$

де C_t – це стан комірки;

h_t – поточний прихований стан.

Пакети τ використовуються для оновлення параметрів мереж акторів та критиків для максимізації $\mathcal{L}(\theta)$ та мінімізації $\mathcal{L}(\varphi)$ за допомогою градієнтного спуску:

$$\tau = [s_t, o_t, a_t, r_t, s_{t+1}, o_{t+1}, a_{t+1}, \ldots] \tag{3.10}$$

Додаємо до мережі LSTM шар, до τ додаються два елементи $h_{t,\pi}$ та $h_{t,V}$ 1, які змінюються на:

$$\tau = [s_t, o_t, h_{t,\pi}, h_{t,V}, a_t, r_t, s_{t+1}, o_{t+1}, h_{t+1,\pi}, h_{t+1,V}, a_{t+1}, \ldots], \tag{3.11}$$

де $h_{t,\pi}$ та $h_{t,V}$ – це приховані стани відповідно мережі акторів та мережі критиків на момент часу t.

Метою є максимізація функції втрат для акторної мережі:

$$\mathcal{L}(\theta) = \frac{1}{Bn} \sum_{i=1}^{B} \sum_{k=1}^{n} \left[\min\left(r_{\theta,i}^k A_i^k, \text{clip}(r_{\theta,i}^k, 1-\epsilon, 1+\epsilon) A_i^k\right) + \sigma \cdot S_\pi \right], \tag{3.12}$$

де $r_{\theta,i}^{k}$ – це співвідношення ймовірностей між новою та старою політиками;

A_i^k – функція переваги;

ϵ – параметр межі відсікання;

S_π – ентропія політики, що стимулює дослідження.

Крім того, у межах дослідження було здійснено спробу визначити функціональне призначення мережі критиків та мережі акторів у багатоагентному алгоритмі навчання з підкріпленням, що базується на архітектурі централізованого навчання з децентралізованим виконанням (CTDE).

Встановлено, що мережа критиків виконує роль центрального контролера, який опрацьовує повний обсяг інформації, отриманої зі спостережень. Це свідчить про те, що додавання шару штучної нейронної мережі до структури мережі критиків теоретично може суттєво підвищити ефективність моделі в умовах часткового спостереження.

Натомість мережа акторів реалізує політику кожного агента окремо. Додавання шару рекурентної нейронної мережі до мережі акторів має менш виражений вплив на її загальну продуктивність.

Отримані експериментальні результати підтверджують обґрунтованість наведеного аналізу, а також доводять доцільність застосування архітектури CTDE для розв’язання даної задачі.

3.7 Багатоцільова оптимізація

Автономне планування маршрутів для групи БПЛА є задачею багатоцільової оптимізації [95]. Необхідно враховувати витрати часу, площу покриття та безпеку системи з декількома БПЛА, як показано на рисунку 3.6. Багатоцільова оптимізація передбачає оптимальний вибір змінних рішення в дискретному просторі рішень.

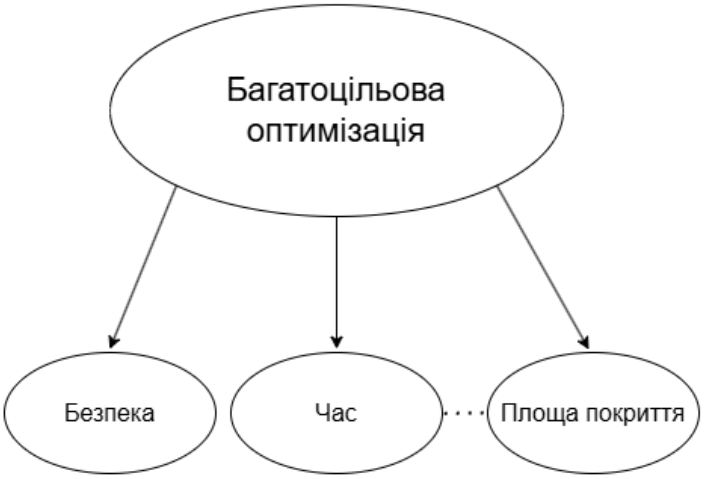


Рисунок 3.6 – Багатоцільова оптимізація

Математично задача багатокритеріальної оптимізації формулюється наступним чином:

$$\max_{\pi} F(\pi) = \max_{\pi} \left[f_1(\pi), f_2(\pi), \ldots, f_m(\pi) \right], \tag{3.13}$$

де m – це кількість критеріїв;

π – стратегія (policy), яка визначає дії системи.

Цей підхід за своєю природою подібний до задачі «вибору дії» у навчанні з підкріпленням. Поєднання «офлайн-навчання» та «онлайн-прийняття рішень» у рамках глибокого навчанні з підкріпленням дозволяє реалізовувати багатоцільову оптимізацію в реальному часі. Таким чином, методи глибокого навчання з підкріпленням є ефективним інструментом для вирішення задач багатоцільової оптимізації в системах з декількома БПЛА, забезпечуючи високу узагальнюваність побудованих моделей.

Функцію винагороди було розроблено з урахуванням принципів багатоцільової оптимізації та попередніх знань у галузі навігації шляхом декомпозиції загальної задачі на низку підзадач. Спільну функцію винагороди сформовано з урахуванням обмежень щодо безпеки, часу виконання та охоплення території. Функція винагороди має вигляд:

$$r_{\text{total}} = \alpha \cdot r_{\text{timecost}} + \beta \cdot r_{\text{security}} + \gamma \cdot r_{\text{coverage}}, \tag{3.14}$$

де $r_{\text{security}} = \sum \left|\text{distance}(UAV_i - UAV_j)\right| - \left|\text{distance}(UAV_i - \text{target}_j)\right|$ – це компонент, що сприяє уникненню зіткнень між БПЛА шляхом контролю відстаней між ними, а також стимулює їх наближення до цільових точок;

$r_{\text{coverage}} = \sum \text{new area}_{UAV_i}$ – винагорода за дослідження нових ділянок території, яка заохочує БПЛА не просто досягати цілей, а й активно обстежувати середовище;

r_{timecost} – компонент, що сприяє мінімізації часу досягнення цільових точок за рахунок скорочення кількості кроків.

Коефіцієнти α, β, γ задаються з урахуванням пріоритетності відповідних критеріїв і можуть змінюватися залежно від вимог до виконання місії. Завдяки цій структурі функції винагороди, система може адаптуватися до різних сценаріїв, зберігаючи при цьому оптимальний баланс між ефективністю, безпекою та охопленням території. Такий підхід дозволяє перетворити багатокритеріальну задачу в однокритеріальну з можливістю подальшого налаштування агресивності стратегії агента без повного перенавчання. Багатоцільову оптимізаційну задачу можна вирішити за допомогою багатоцільового навчання з підкріпленням, як показано на рисунку 3.7.

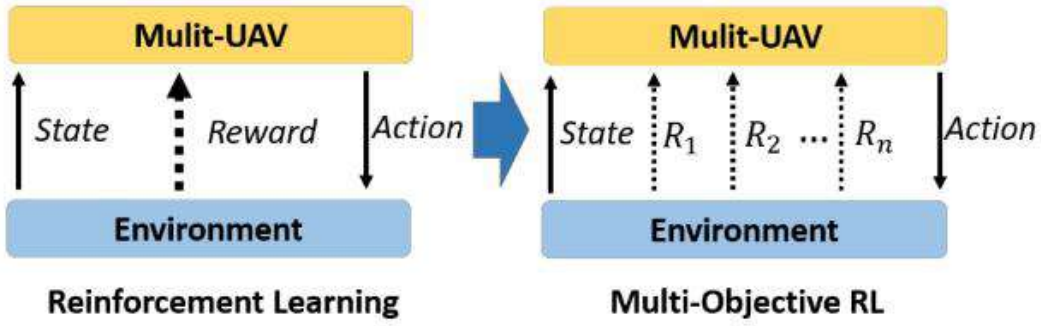


Рисунок 3.7 – Різниця між навчанням з підкріпленням та багатоцільовим навчанням з підкріпленням [96]

Набір рішень, який називається «фронт Парето», може представляти оптимальні рішення усіма різними вагами. Відмінності між двома методами показані на рисунку 3.8.

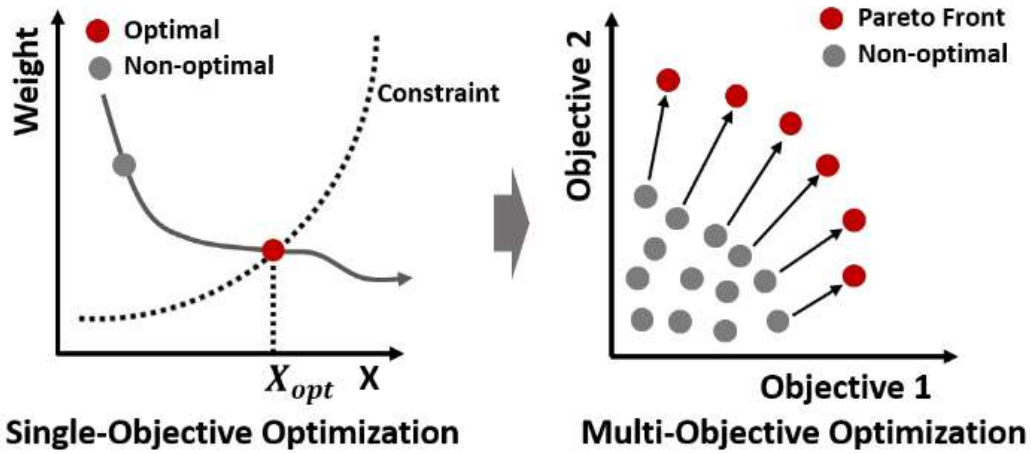


Рисунок 3.8 – Багатоцільова оптимізація: отримання оптимального рішення при обмеженнях безпеки, часу та покриття [97]

Для вирішення задачі багатокритеріальної оптимізації політики використовується градієнт багатокритеріального підходу. Такий підхід спрямований на максимізацію зваженої суми винагород, де вектор ваг ω визначає важливість кожної з цільових функцій. Це призводить до зміни вигляду градієнта політики.

Функціонал мети багатокритеріальної політики можна представити у вигляді:

$$\mathcal{J}(\theta,\omega) = \omega^{\top} F(\pi) = \sum_{i=1}^{m} \omega_i f_i(\pi) = \sum_{i=1}^{m} \omega_i \mathcal{J}_i^{\pi}, \tag{3.15}$$

де θ – це параметри політики;

$\omega \in \mathbb{R}^m$ – вектор ваг для кожної цілі;

$\mathcal{J}_i^{\pi}$ – функціонал мети для i-ої задачі.

Градієнт цієї функції за параметрами політики має вигляд:

$$\nabla_\theta \mathcal{J}(\theta, \omega) = \sum_{i=1}^{m} \omega_i \nabla_\theta \mathcal{J}_i(\theta) \tag{3.16}$$

Що далі можна подати у вигляді математичного сподівання:

$$\begin{aligned} &= \mathbb{E}\left[\sum_{t=0}^{T} \omega^\top A^\pi(s_t, a_t) \nabla_\theta \pi_\theta(a_t | s_t) \right] \\ &= \mathbb{E}\left[\sum_{t=0}^{T} A_\omega^\pi(s_t, a_t) \nabla_\theta \pi_\theta(a_t | s_t) \right], \end{aligned} \tag{3.17}$$

де A^π – це перевага (advantage function);

A_ω^π – зважена перевага, яка враховує всі цілі з відповідними вагами.

3.8 Висновки до третього розділу

В даному розділі було вирішено проблему, пов'язану з плануванням траєкторії руху декількох БПЛА з неповною інформацією, за допомогою навчання з підкріпленням на основі централізованого навчання та децентралізованої архітектури виконання. Виконано глибоке дослідження причини, чому централізоване навчання та децентралізована архітектура виконання покращують продуктивність моделі, та поясненно переваги централізованого навчання порівняно з повністю розподіленими методами.

При розробці функції винагороди було застосовано принцип декомпозиції, що дозволило ефективно вирішити багатоцільову оптимізаційну задачу з урахуванням критеріїв безпеки, часу виконання та охоплення території. Функція винагороди побудована таким чином, щоб забезпечити гнучке балансування між різними цілями місії.

Також реалізовано підхід на основі алгоритму проксимальної оптимізації політики (PPO) у поєднанні з архітектурою централізованого навчання та

децентралізованого виконання (CTDE), що дозволяє дронам ефективно навчатися в умовах часткової спостережуваності середовища. Додано рекурентний шар (RNN, зокрема LSTM) в структуру мереж, що дозволило зменшити вплив неповної інформації, покращити узагальнення та стабільність навчання.

Підсумовуючи, запропонований підхід дозволяє створити стійку багатоагентну систему автономного планування маршрутів для групи БПЛА, яка демонструє хорошу адаптивність до складних умов середовища, приймає рішення в реальному часі та забезпечує ефективне досягнення спільної мети з урахуванням множини суперечливих критеріїв.

4 МОДЕЛЮВАННЯ СИСТЕМИ ПЛАНУВАННЯ РУХУ ДЛЯ ГРУПИ БПЛА

4.1 Архітектура мережі

У цьому дослідженні для побудови трирівневої нейронної мережі актора та критика алгоритму проксимальної оптимізації політики (PPO) було використано бібліотеку PyTorch [98]. Для координації дій усіх безпілотних літальних апаратів була реалізована парадигма централізованого навчання з децентралізованим виконанням (CTDE). Інтуїтивна відмінність між централізованим та незалежним навчанням полягає у складі вхідних даних для мережі критика.

У рамках експерименту локальні спостереження усіх БПЛА було агреговано у вектор великої розмірності, що сформував спільне спостереження. Цей вектор подавався на вхід до мережі критика, яку позначено як O_{center}. Водночас на вхід до мережі актора подавалося індивідуальне спостереження O_i кожного БПЛА, відповідно до схеми, представленої на рисунку 4.1.

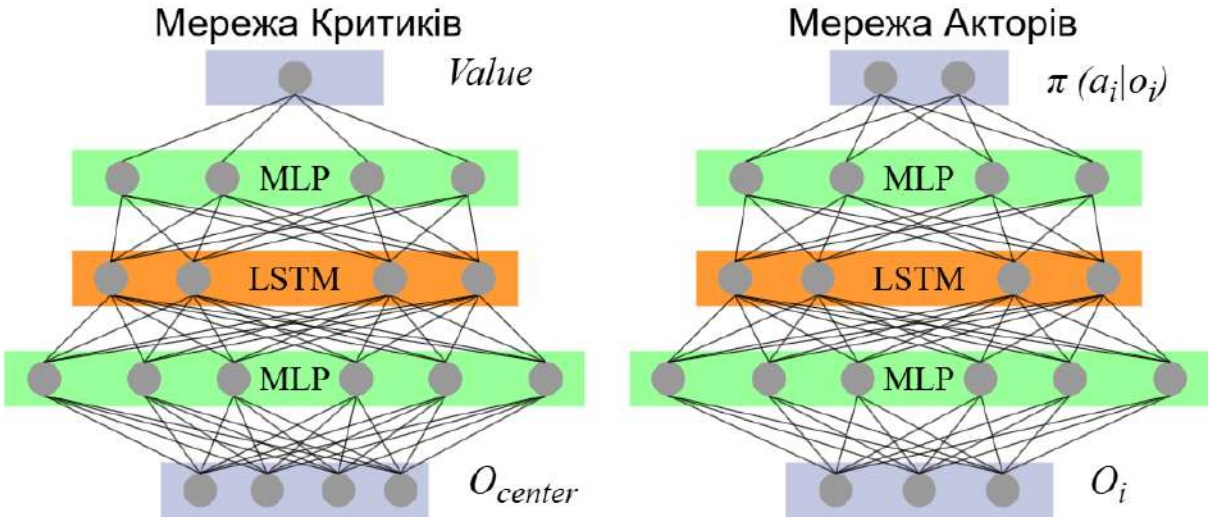


Рисунок 4.1 – Мережа акторів та мережа критиків, а також всі агенти, які поділяють ці мережі

З метою оцінки ефективності підходу CTDE порівняно з незалежним навчанням було проведено чотири контрольні експерименти. Архітектура мережі

складалася з трьох шарів: перший і третій – повнозв’язані, другий – рекурентний шар типу LSTM.

Для верифікації доцільності використання LSTM-шару, було створено варіант тієї самої архітектури без рекурентного шару. Це дозволило провести порівняльний аналіз із метою визначення, який з компонентів – мережа критика чи оцінки цінності – у більшій мірі залежить від історичних даних, тим самим підтверджуючи переваги застосування CTDE-архітектури.

Проксимальна оптимізація політики виводить випадкові політики, що означає, що виходи мережі акторів – це μ, σ, які є математичним сподіванням і дисперсією гаусівського розподілу, а вихідна дія є випадковою вибіркою з цього гаусівського розподілу.

У процесі тренування агентів методом PPO використовувалися фіксовані гіперпараметри, наведені в таблиці 4.1. Тренування відбувалося протягом 625 епізодів, кожен з яких мав максимальну довжину 200 кроків. Для ефективного збору досвіду застосовувалося 16 паралельних потоків розгортання середовища. Параметр «clipping» зі значенням 0.2 обмежував оновлення політики, запобігаючи надто різким змінам, що є характерною особливістю PPO. Коефіцієнт дисконтування γ дорівнював 0.99, що означає високий ступінь урахування майбутніх нагород. Для збереження балансу між дослідженням нових дій і використанням вже вивчених стратегій застосовувався коефіцієнт ентропії 0.1.

Зібрані дані зберігалися у буфері розміром 500 кроків, а під час оновлення мережі використовувалися мініпакети (batch) по 32 зразки. Архітектура нейронної мережі включала повнозв’язний шар (Fully Connected) розміром 128 нейронів і рекурентний шар із прихованим станом розміром 64. В якості функції активації використовувалася ReLU (Rectified Linear Unit), яка пропускає позитивні значення, а негативні перетворює на нуль, що забезпечує ефективне навчання та зменшує ризик затухаючих градієнтів. Для оптимізації ваг нейронної мережі застосовувався алгоритм Adam – адаптивний оптимізатор, який поєднує метод моменту та адаптивне масштабування градієнтів, забезпечуючи швидку й стабільну збіжність.

Таблиця 4.1 – Параметри нейронної мережі та тренування PPO-агента

№ п.п	Параметр	Значення
1	Епізод	625
2	Довжина епізоду	200
3	Потоки розгортання	16
4	Обмеження (clip)	0,2
5	Коефіцієнт дисконтування	0,99
6	Коефіцієнт ентропії	0,1
7	Буфер (buffer size)	500
8	Пакет (batch size)	32
9	Повнозв’язний шар	128
10	Рекурентний шар	64
11	Функція активації	ReLU
12	Оптимізатор	Adam

4.2 Проведення моделювання та тестування

Після виконання одного мільйона кроків навчання та аналізу результатів експерименту було встановлено, що алгоритм PPO, реалізований на основі централізованого навчання з децентралізованим виконанням, продемонстрував вищу ефективність порівняно з незалежним навчанням у задачах автономного планування траєкторії для групи БПЛА.

Як видно з результатів, наведених на рисунку 4.2, повністю незалежним і розподіленим методам навчання складно забезпечити належну продуктивність у задачах із кількома БПЛА. Запровадження архітектури CTDE суттєво підвищило ефективність: значення функції винагороди стало додатним, а продуктивність зросла навіть зі збільшенням кількості БПЛА.

Це підтверджує доцільність використання архітектури CTDE для розв’язання розподілених завдань. Центральний контролер забезпечує координацію дій усіх БПЛА, однак не визначає обмежень щодо їх кількості. Водночас,

експериментально встановлено, що за умови використання більше ніж шести БПЛА центральний контролер втрачає ефективність через надмірну розмірність простору спільних спостережень.

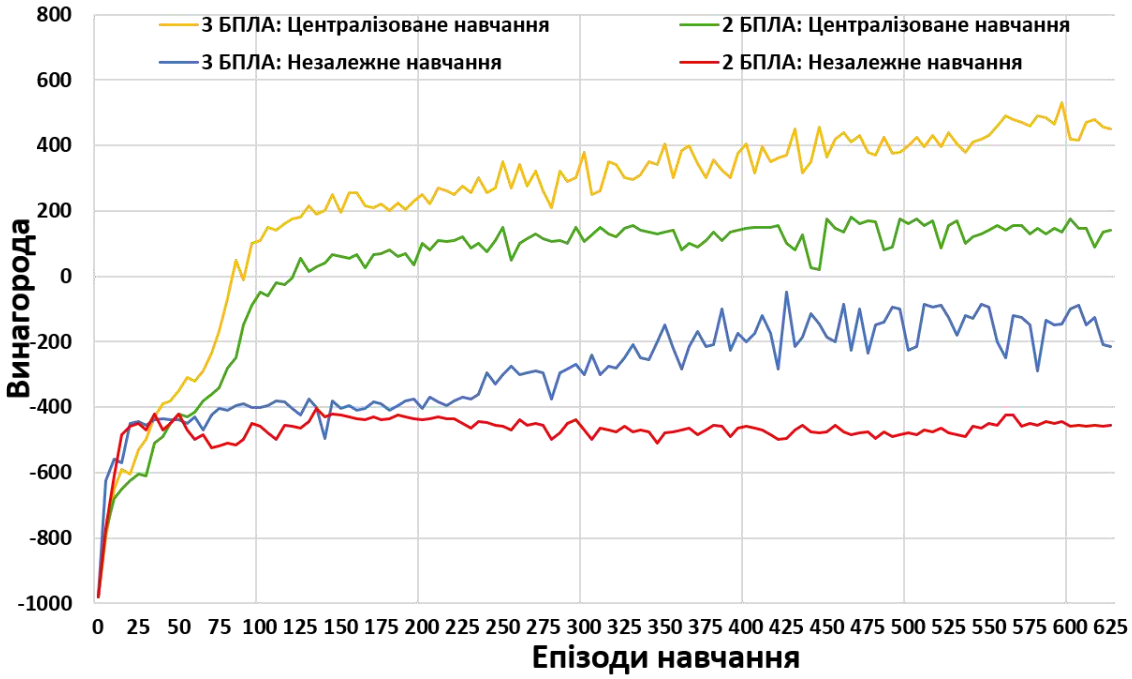


Рисунок 4.2 – Порівняння CTDE та незалежної архітектури

Крім того, було проведено низку контрольних експериментів з метою перевірки, чи може додавання шару рекурентної нейронної мережі розв'язати проблему складнощів навчання багатьох БПЛА в умовах неповної інформації.

Відповідно до результатів експериментів, наведених на рисунку 4.3, встановлено, що додавання шару RNN до мереж як акторів, так і критиків суттєво підвищує ефективність моделі. Зокрема, включення шару RNN до мережі критиків забезпечило подібний рівень покращення, однак процес збіжності відбувався повільніше. Натомість додавання шару RNN лише до мережі акторів не дало істотного приросту продуктивності моделі та не усунуло проблем, пов'язаних із частковими спостереженнями у випадку з декількома БПЛА.

Отримані результати підтверджують наш аналітичний висновок: у межах архітектури CTDE мережа критиків виконує функцію центрального контролера, що координує дії всіх БПЛА задля досягнення спільної мети, опрацьовуючи спільні

спостереження. Додавання шару RNN до мережі критиків виявилося ефективним рішенням, оскільки проблема неповної інформації компенсується за рахунок використання прихованого стану.

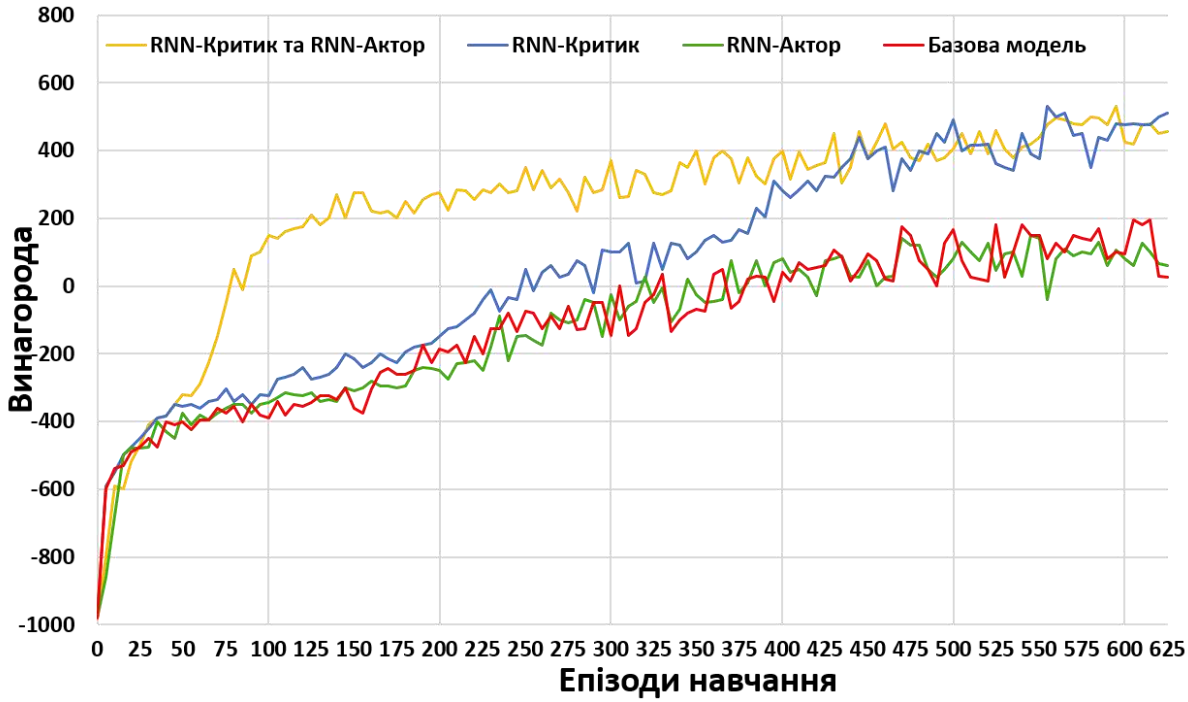


Рисунок 4.3 – Ефективність моделі після додавання шару RNN до мережі акторів і критиків

Для того, щоб вирішити проблему прийняття рішень з неповною інформацією, було обрано архітектуру CTDE і додано шар RNN для використання історичної інформації. В алгоритмах безмодельного навчання з підкріпленням винагорода є важливим критерієм оцінки ефективності. Аналогічно, втрата цінності, втрата політики та ентропія дії також є ключовими компонентами для оцінки продуктивності алгоритму. Втрата цінності дає змогу оцінити вихідну цінність мережі-критика і визначає, чи є прогноз точним, а ентропія дії відображає випадковість виходу стратегії мережі-актора. Тут були сподівання, що ентропія дії буде достатньо великою, щоб сприяти адекватному дослідженню.

Результати експерименту доводять, що алгоритм значно покращив продуктивність. Як показано на рисунку 4.4, після приблизно 300 епізодів функція втрат починає стабілізуватися, а швидка збіжність значень втрат також показала,

що значення, передбачене критичною мережею, було більш точним. Аналогічно, після додавання шару LSTM втрати критиків зменшилися, а значення ентропії політики плавно знижувалося, що було хорошим показником і означало, що агенти не потрапляли в локальний оптимум. Для дослідження було не потрібно, щоб це значення спадало надто швидко або надто повільно. Ентропія політики – це дисперсія вихідних дій, і плавна крива показує, що група безпілотників засвоїла стабільну політику після достатнього дослідження, оскільки дослідження є незамінним у навчанні з підкріпленням.

Крім того, додавання шару LSTM суттєво покращило загальну продуктивність алгоритму. Середнє значення винагороди помітно зросло, що свідчить про здатність агента приймати більш обґрунтовані рішення. Водночас ентропія політики та функція втрат критика збігалися швидше, що засвідчує ефективність застосування історичної інформації через мережу LSTM. Параметри моделі динамічно оновлювалися протягом усього процесу навчання.

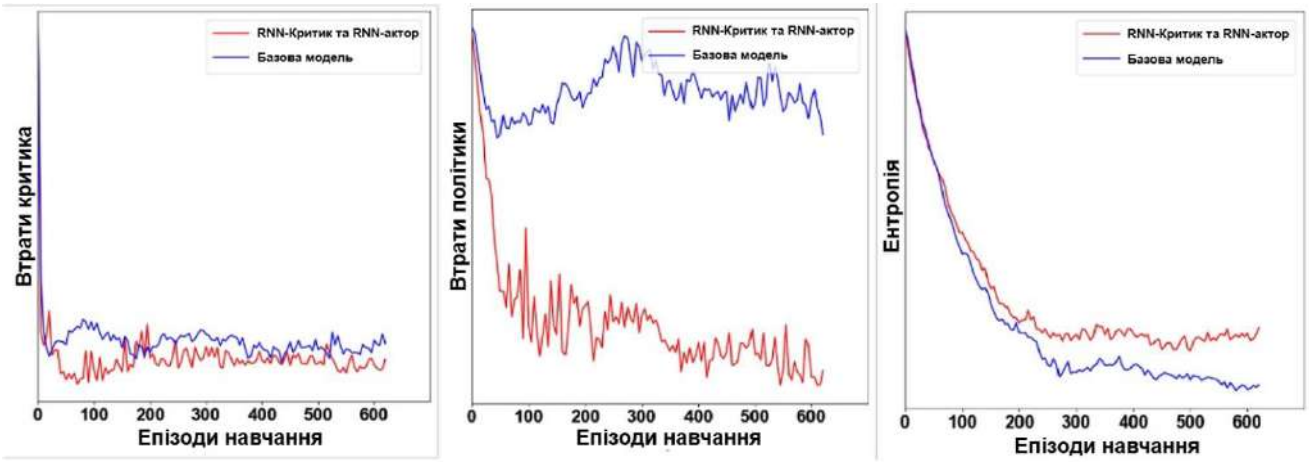


Рисунок 4.4 – Зміна функцій втрат критика, політики та ентропії протягом тренувальних епізодів

Навчання з підкріпленням демонструє значно вищу швидкість виконання порівняно з традиційними алгоритмами ройового інтелекту, що робить його придатним для вирішення задач прийняття рішень у реальному часі (див. рисунок 4.5). Після завершення етапу навчання параметри вагових коефіцієнтів нейронної

мережі було зафіксовано (заморожено), що забезпечило стабільність поведінки моделі під час виконання. При розв’язанні навігаційної задачі за допомогою навченої моделі середнє значення винагороди, досягнуте запропонованим методом, було вищим і стабільнішим порівняно з альтернативними підходами.

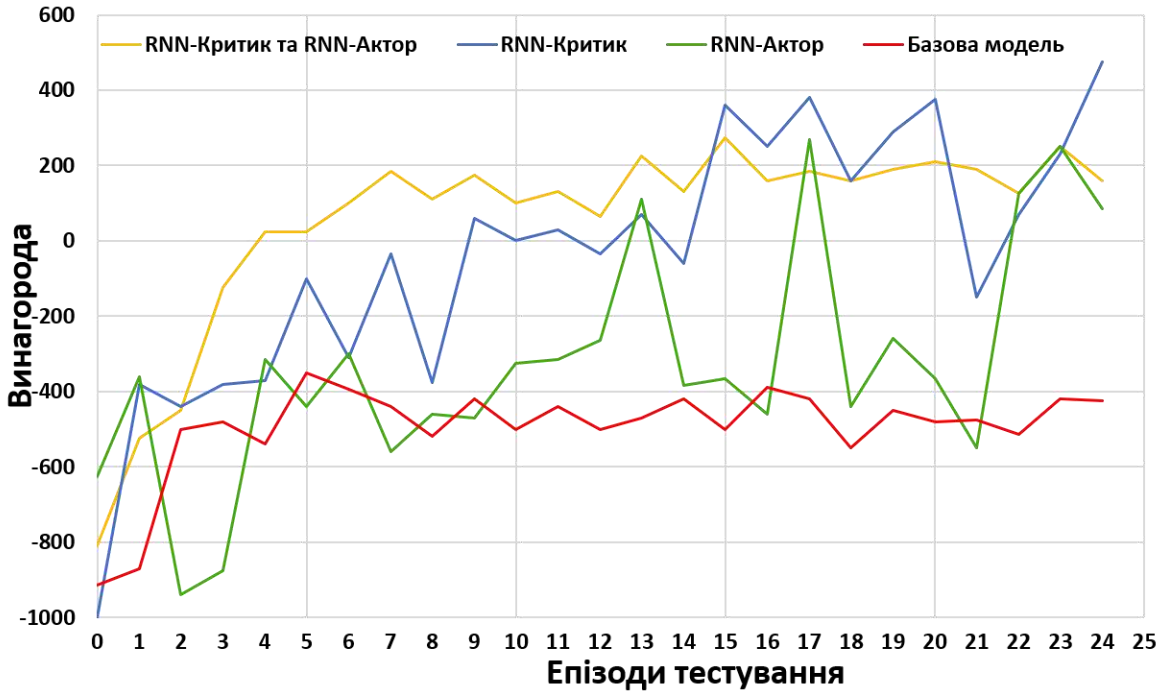


Рисунок 4.5 – Порівняння середньої винагороди різних архітектур на етапі тестування

Після завершення етапу навчання модель була використана для онлайн-планування траєкторії для групи БПЛА у тривимірному просторі. Це планування здійснювалося на етапі інференції з використанням попередньо навчених параметрів. Як показано на рисунку 4.6, розроблений підхід успішно забезпечив побудову маршрутів у режимі реального часу з урахуванням обмежень щодо безпеки, часу та покриття.

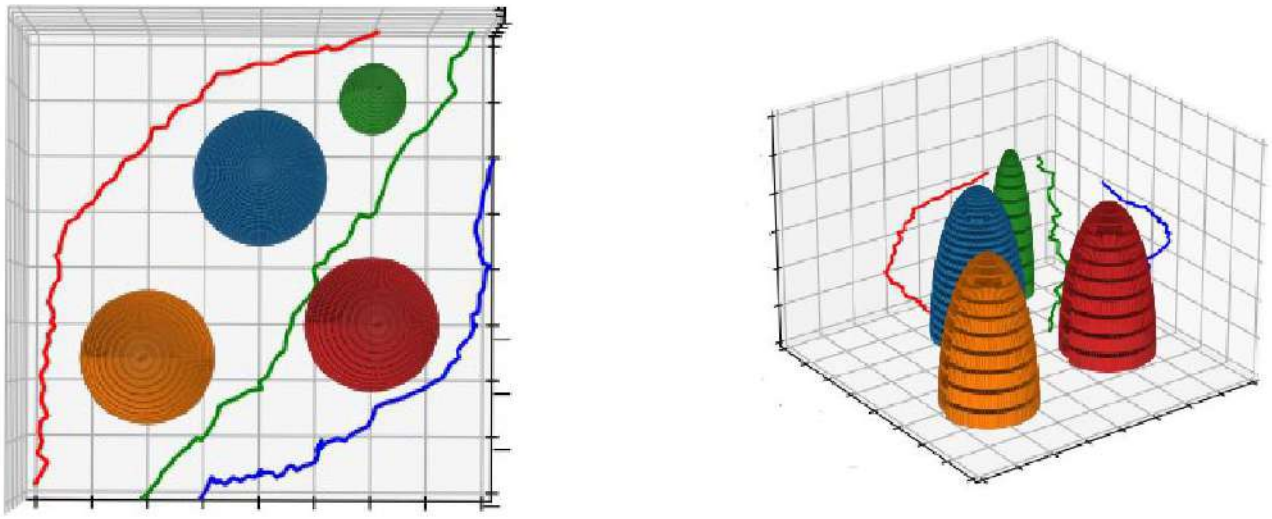


Рисунок 4.6 – Планування траєкторії руху декількох БПЛА в тривимірному середовищі за допомогою навчання з підкріпленням

У порівнянні з традиційними алгоритмами оптимізації, такими як алгоритм роїв частинок, як показано в таблиці 4.2 запропонований метод демонструє кращі показники, зокрема в аспектах стабільності, швидкості реагування та точності прокладення маршруту. Це підкреслює потенціал навчання з підкріпленням для ефективного розв’язання задач побудови траєкторій у складному динамічному середовищі.

Таблиця 4.2 – Порівняння продуктивності запропонованого методу з іншими сучасними алгоритмами

Метод	Витрати часу	Покриття	Відсоток успішності	Опорна швидкість, с
Запропонований метод	92	57.9%	92.7%	0.126
Навчання з підкріпленням	93	54.5%	90.1%	0.115
DPSO	97	41.2%	65.2%	1.35
GAPSO	95	43.7%	62.1%	1.16

4.3 Висновки до четвертого розділу

У цьому розділі було проведено моделювання та емпіричну оцінку ефективності системи планування руху для групи безпілотних літальних апаратів (БПЛА) з використанням навчання з підкріпленням на основі алгоритму проксимальної оптимізації політики (PPO) в архітектурі централізованого навчання з децентралізованим виконанням (CTDE).

Побудована нейронна мережа складалася з трьох основних рівнів, де використовувались як повнозв’язані шари, так і рекурентний шар LSTM, що дозволило враховувати історичну інформацію при прийнятті рішень. Була реалізована чітка розділеність між мережею критиків, яка отримувала агреговані спостереження (спільне бачення середовища), і мережею акторів, що базувалася на локальних спостереженнях кожного агента. Це дозволило максимально ефективно використати переваги парадигми CTDE.

Проведено серію експериментів, які підтвердили, що підхід CTDE значно перевершує традиційні незалежні архітектури навчання, особливо в умовах неповної інформації та складної багатокомпонентної взаємодії між агентами. Застосування LSTM-шару в мережі критика дозволило суттєво знизити функцію втрат та покращити узгодженість дій між агентами. Водночас було встановлено, що додавання рекурентного шару до мережі акторів мало обмежений вплив на загальну продуктивність системи, що свідчить про ключову роль критика як центрального координатора.

Особливу увагу було приділено динаміці змін функцій втрат та ентропії дії протягом процесу навчання. Стабілізація цих показників після певної кількості епізодів свідчить про успішне засвоєння ефективної політики дій агентами. Плавне зменшення ентропії продемонструвало поступове формування збалансованої стратегії, здатної поєднувати як дослідження, так і експлуатацію вивчених рішень.

Отримані результати також засвідчили переваги запропонованого підходу в порівнянні з альтернативними методами, такими як алгоритми ройового інтелекту (DPSO, GAPSO). Зокрема, було досягнуто вищого рівня стабільності, покриття

території та швидкості прийняття рішень, що особливо важливо у сценаріях реального часу та підвищених вимог до безпеки. На основі таблиці порівняння видно, що запропонований метод забезпечує найкращий баланс між витратами часу, точністю, ефективністю та успішністю місій.

Фінальне моделювання показало, що система на основі попередньо навчених нейромереж здатна здійснювати онлайн-планування траєкторії у тривимірному просторі з урахуванням обмежень, пов’язаних із безпекою, часовими ресурсами та необхідністю покриття максимальної площі. Таким чином, впровадження навчання з підкріпленням у кооперативне планування дій БПЛА дозволяє значно розширити функціональні можливості автономних систем у складних, динамічних середовищах.

ВИСНОВКИ

У роботі за результатами виконаних теоретичних та практичних досліджень розроблено систему планування та моніторингу руху групи БПЛА на основі аналізу параметрів їх компонентів, що враховує умови неповної інформації та динаміку зовнішнього середовища. Запропонована система поєднує алгоритми навчання з підкріпленням, архітектуру централізованого навчання з децентралізованим виконанням (CTDE) та рекурентні нейронні мережі для забезпечення адаптивного керування груповим польотом.

У першому розділі проведено огляд характеристик, компонентів та сфер застосування БПЛА, проаналізовано сучасні методи енергозабезпечення та підзарядки, а також вивчено актуальні підходи до планування і моніторингу групових польотів. Визначено основні технічні обмеження й сформульовано постановку задачі дослідження.

У другому розділі розглянуто теоретичні основи машинного навчання, зокрема контрольоване, неконтрольоване та навчання з підкріпленням. Проведено порівняльний аналіз алгоритмів з позиції їх застосування у задачах планування маршруту для груп БПЛА.

У третьому розділі розроблено математичну модель планування траєкторій в умовах неповної інформації. Реалізовано та вдосконалено алгоритм проксимальної оптимізації політики (PPO), заснований на акторно-критичній архітектурі з використанням CTDE. Для обробки історичних даних та компенсації часткової спостережуваності додано шар рекурентної нейронної мережі (RNN). Також сформульовано функцію багатоцільової винагороди.

У четвертому розділі описано архітектуру розробленої системи та проведено тестування її роботи в симульованому середовищі. Проведена емпірична оцінка ефективності показала, що система здатна адаптуватися до змінних умов і забезпечує стабільне планування та моніторинг руху групи БПЛА.

Набула подальшого розвитку методика планування та моніторингу руху групи БПЛА в умовах неповної інформації із використанням навчання з

підкріпленням. Удосконалено архітектуру PPO-алгоритму завдяки інтеграції рекурентних нейронних мереж та застосуванню централізованого навчання з децентралізованим виконанням, що підвищило ефективність координації та адаптивності системи в динамічному середовищі.

Перспективи подальших досліджень полягають у розширенні функціональності системи за рахунок інтеграції додаткових факторів (наприклад, погодні умови, обмеження зони польоту, взаємодія з наземними об'єктами), а також використанні більш складних моделей прогнозування стану середовища на основі комбінації навчання з підкріпленням і комп’ютерного зору.

За темою кваліфікаційної роботи опубліковано одну статтю [99] у фаховому науковому журналі: VELYCHKO M., KYSIL T. M. REINFORCEMENT LEARNING METHOD FOR AUTONOMOUS FLIGHT PATH PLANNING OF MULTIPLE UAVS. *Computer Systems and Information Technologies*. 2025. №. 2.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Overview of Propulsion Systems for Unmanned Aerial Vehicles / B. Zhang et al. *Energies*. 2022. Vol. 15, no. 2. P. 455. DOI: https://doi.org/10.3390/en15020455.

2. Єршова О. Л., Бажан Л. І. Розумне місто: концепція, моделі, технології, стандартизація. *Статистика України*. 2020. № 2–3. С. 68–77. DOI: 10.31767/su.2-3(89-90)2020.02-03 08.

3. A joint global and local path planning optimization for UAV task scheduling towards crowd air monitoring / Y. Tang et al. *Computer Networks*. 2021. Vol. 193. P. 107913. DOI: https://doi.org/10.1016/j.comnet.2021.107913.

4. Drone Routing for Drone-Based Delivery Systems: A Review of Trajectory Planning, Charging, and Security / A. M. Raivi et al. *Sensors*. 2023. Vol. 23, no. 3. P. 1463. DOI: https://doi.org/10.3390/s23031463.

5. The Employment of Unmanned Aerial Vehicles for Analyzing and Mitigating Disaster Risks in Industrial Sites / G. Aiello et al. *IEEE Transactions on Engineering Management*. 2020. Vol. 67, no. 3. P. 519–530. DOI: https://doi.org/10.1109/tem.2019.2949479.

6. Optimization approaches for civil applications of unmanned aerial vehicles (UAVs) or aerial drones: A survey / A. Otto et al. *Networks*. 2018. Vol. 72, no. 4. P. 411–458. DOI: https://doi.org/10.1002/net.21818.

7. Ivanova S., Prosekov A., Kaledin A. A Survey on Monitoring of Wild Animals during Fires Using Drones. *Fire*. 2022. Vol. 5, no. 3. P. 60. DOI: https://doi.org/10.3390/fire5030060 (date of access: 16.04.2025).

8. Quantifying the movement, behaviour and environmental context of group-living animals using drones and computer vision / B. Koger et al. *Journal of Animal Ecology*. 2023. DOI: https://doi.org/10.1111/1365-2656.13904.

9. UAV remote sensing applications in marine monitoring: Knowledge visualization and review / Z. Yang et al. *Science of The Total Environment*. 2022. P. 155939. DOI: https://doi.org/10.1016/j.scitotenv.2022.155939.

10.Elmokadem T., Savkin A. V. Towards Fully Autonomous UAVs: A Survey. *Sensors*. 2021. Vol. 21, no. 18. P. 6223. DOI: https://doi.org/10.3390/s21186223.

11.Lee D., Hess D. J., Heldeweg M. A. Safety and privacy regulations for unmanned aerial vehicles: A multiple comparative analysis. *Technology in Society*. 2022. Vol. 71. P. 102079. DOI: https://doi.org/10.1016/j.techsoc.2022.102079.

12.A survey on security and privacy issues of UAVs / Y. Mekdad et al. *Computer Networks*. 2023. P. 109626. DOI: https://doi.org/10.1016/j.comnet.2023.109626.

13.Xing L., Johnson B. W. Reliability Theory and Practice for Unmanned Aerial Vehicles. *IEEE Internet of Things Journal*. 2022. P. 1. DOI: https://doi.org/10.1109/jiot.2022.3218491.

14.Frattolillo F., Brunori D., Iocchi L. Scalable and Cooperative Deep Reinforcement Learning Approaches for Multi-UAV Systems: A Systematic Review. *Drones*. 2023. Vol. 7, no. 4. P. 236. DOI: https://doi.org/10.3390/drones7040236.

15.Khan A., Gupta S., Gupta S. K. Emerging UAV technology for disaster detection, mitigation, response, and preparedness. *Journal of Field Robotics*. 2022. DOI: https://doi.org/10.1002/rob.22075.

16.Unmanned Aerial Vehicles for Search and Rescue: A Survey / M. Lyu et al. *Remote Sensing*. 2023. Vol. 15, no. 13. P. 3266. DOI: https://doi.org/10.3390/rs15133266.

17.Lekidis A., Anastasiadis A. G., Vokas G. A. Electricity infrastructure inspection using AI and edge platform-based UAVs. *Energy Reports*. 2022. Vol. 8. P. 1394–1411. DOI: https://doi.org/10.1016/j.egyr.2022.07.115.

18. A Distributed Control Framework of Multiple Unmanned Aerial Vehicles for Dynamic Wildfire Tracking / H. X. Pham et al. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*. 2020. Vol. 50, no. 4. P. 1537–1548. DOI: https://doi.org/10.1109/tsmc.2018.2815988.

19.Exploring the transformative role of drone technology in advancing healthcare delivery in Africa; a perspective / G. Olatunji et al. *Annals of Medicine & Surgery*. 2023. Vol. 85, no. 10. P. 5279–5284. DOI: https://doi.org/10.1097/ms9.0000000000001221.

20.Emad Alfaris R., Vafakhah Z., Jalayer M. Application of Drones in Humanitarian Relief: A Review of State of Art and Recent Advances and Recommendations. *Transportation Research Record: Journal of the Transportation Research Board*. 2023. DOI: https://doi.org/10.1177/03611981231209033.

21.Airborne! UAV Delivery of Blood Products and Medical Logistics for Combat Zones / D. T. Lammers et al. *Transfusion*. 2023. DOI: https://doi.org/10.1111/trf.17329.

22.Da Silva I. D., Caillouet C. Optimizing the trajectory of drones: trade-off between distance and energy. *2020 IEEE International Conference on Sensing, Communication and Networking (SECON Workshops)*, Como, Italy, 22–26 June 2020. 2020. DOI: https://doi.org/10.1109/seconworkshops50264.2020.9149781.

23.Multi-UAV Autonomous Path Planning in Reconnaissance Missions Considering Incomplete Information: A Reinforcement Learning Method / Y. Chen et al. *Drones*. 2022. Vol. 7, no. 1. P. 10. DOI: https://doi.org/10.3390/drones7010010.

24.Rechargeable UAV Trajectory Optimization for Real-Time Persistent Data Collection of Large-Scale Sensor Networks / R. Wang et al. *IEEE Transactions on Communications*. 2024. P. 1. DOI: https://doi.org/10.1109/tcomm.2024.3493812.

25.UAV Fleet Based Accident Monitoring Systems with Automatic Battery Replacement Systems: Algorithms for Justifying Composition and Use Planning / I. Kliushnikov et al. *International Journal of Safety and Security Engineering*. 2021. Vol. 11, no. 4. P. 319–328. DOI: https://doi.org/10.18280/ijsse.110404.

26.Tachinina O., Lysenko A., Kutiepov V. Classification of Modern Unmanned Aerial Vehicles. *Electronics and Control Systems*. 2022. Vol. 4, no. 74. P. 79–86. DOI: https://doi.org/10.18372/1990-5548.74.17354.

27.Drone delivery operations underway in 27 countries - Unmanned airspace. Unmanned airspace. URL: https://www.unmannedairspace.info/latest-news-and-information/drone-delivery-operations-underway-in-26-countries/ (дата звернення: 25.02.2025).

28.Drone Routing for Drone-Based Delivery Systems: A Review of Trajectory Planning, Charging, and Security / A. M. Raivi et al. *Sensors*. 2023. Vol. 23, no. 3. P. 1463. DOI: https://doi.org/10.3390/s23031463.

29.Classification of the Unmanned Aerial Systems. *GEOG 892: Unmanned Aerial Systems*. URL: https://www.e-education.psu.edu/geog892/node/5 (дата звернення: 25.02.2025).

30.Dias da Silva I. Optimization of UAVs deployment and coordination for exploration and monitoring applications : Thesis. 2023. URL: https://theses.hal.science/tel-04257344v1.

31.Yildirim Ş., İşci M. Performance analysis of multi rotor drone systems with changeable rotors. *International Review of Applied Sciences and Engineering*. 2022. DOI: https://doi.org/10.1556/1848.2021.00413.

32.Garg P. K. Characterisation of Fixed-Wing Versus Multirotors UAVs/Drones. *Journal of Geomatics*. 2022. Vol. 16, no. 2. P. 152–159. DOI: https://doi.org/10.58825/jog.2022.16.2.44.

33.Drone Development from Concept to Flight: Design, Assemble, and Discover the Applications of Unmanned Aerial Vehicles. Packt Publishing, Limited, 2024.

34.Yoon J., Doh J. Optimal PID control for hovering stabilization of quadcopter using long short term memory. *Advanced Engineering Informatics*. 2022. Vol. 53. P. 101679. DOI: https://doi.org/10.1016/j.aei.2022.101679.

35.Chittoor P. K., Chokkalingam B., Mihet-Popa L. A Review on UAV Wireless Charging: Fundamentals, Applications, Charging Techniques and Standards. *IEEE Access*. 2021. Vol. 9. P. 69235–69266. DOI: https://doi.org/10.1109/access.2021.3077041.

36.Bae C.-H., Kim C.-H., Lee S. A Ground Control System Based on Digital Twin for Monitoring the Status of UAVs. *CEUR Workshop Proceedings : Proceedings of the International Scientific Conference, Seoul, 4 December 2023*. 2023. URL: https://ceur-ws.org/Vol-3655/ISE2023_10_Bae_A_Ground.pdf.

37.A Comprehensive Review of Micro UAV Charging Techniques / S. A. H. Mohsan et al. *Micromachines*. 2022. Vol. 13, no. 6. P. 977.

38.A Comprehensive Review on Wireless Power Transfer Systems for Charging Portable Electronics / A. Laha et al. *Eng*. 2023. Vol. 4, no. 2. P. 1023–1057.

39.A Comprehensive Review on Wireless Capacitive Power Transfer Technology: Fundamentals and Applications / M. Z. Erel et al. *IEEE Access*. 2022. Vol. 10. P. 3116–3143.

40.Wang Y., Han Z. Ant colony optimization for traveling salesman problem based on parameters optimization. *Applied Soft Computing*. 2021. Vol. 107. P. 107439. DOI: https://doi.org/10.1016/j.asoc.2021.107439.

41.A Secure Communication Protocol for Unmanned Aerial Vehicles / N. Ali Khan et al. *Computers, Materials & Continua*. 2022. Vol. 70, no. 1. P. 601–618. DOI: https://doi.org/10.32604/cmc.2022.019419.

42.Caillouet C., Giroire F., Razafindralambo T. Efficient data collection and tracking with flying drones. *Ad Hoc Networks*. 2019. Vol. 89. P. 35–46.

43.Digital twin modeling / F. Tao et al. *Journal of Manufacturing Systems*. 2022. Vol. 64. P. 372–389. DOI: https://doi.org/10.1016/j.jmsy.2022.06.015.

44.Challenges of Ground Control System in Ensuring Safe Flights for Unmanned Aerial Vehicles / A. Mohammed et al. *MEKATRONIKA*. 2022. Vol. 4, no. 1. P. 8–19. DOI: https://doi.org/10.15282/mekatronika.v4i1.7350.

45.What is machine learning? URL: https://www.ibm.com/think/topics/machine-learning (дата звернення 10.03.2025).

46.Supervised versus unsupervised learning: What's the difference? URL: https://www.ibm.com/think/topics/supervised-vs-unsupervised-learning (дата звернення 10.03.2025).

47.Supervised vs Unsupervised Learning: Differences, Applications, and Market Trends. URL: https://eleks.com/blog/supervised-unsupervised-learning/ (дата звернення 10.03.2025).

48.What Is Logistic Regression? Equation, Assumptions, Types, and Best Practices. URL: https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-logistic-regression/ (дата звернення 11.03.2025).

49.Sigmoid Function. URL: https://www.geeksforgeeks.org/derivative-of-the-sigmoid-function/ (дата звернення 11.03.2025).

50.Decision Tree. URL: https://www.geeksforgeeks.org/decision-tree/ (дата звернення 11.03.2025).

51.Support Vector Machine (SVM) Algorithm. URL: https://www.geeksforgeeks.org/support-vector-machine-algorithm/ (дата звернення 11.03.2025).

52.K-Nearest Neighbor(KNN) Algorithm. URL: https://www.geeksforgeeks.org/k-nearest-neighbours/ (дата звернення 11.03.2025).

53.What is the k-nearest neighbors (KNN) algorithm? URL: https://www.ibm.com/think/topics/knn (дата звернення 11.03.2025).

54.Naive Bayes Classifiers. URL: https://www.geeksforgeeks.org/naive-bayes-classifiers/ (дата звернення 12.03.2025).

55.A Practical Guide in Implementing Naïve Bayes using Python. URL: https://medium.com/@didulathavishaplus/a-practical-guide-in-implementing-na%C3%AFve-bayes-using-python-005e805e7a91 (дата звернення 12.03.2025).

56.What is linear regression? URL: https://www.ibm.com/think/topics/linear-regression (дата звернення 12.03.2025).

57.Linear Regression in Machine learning. URL: https://www.geeksforgeeks.org/ml-linear-regression/ (дата звернення 12.03.2025).

58.Implementation of Polynomial Regression. URL: https://www.geeksforgeeks.org/python-implementation-of-polynomial-regression/ (дата звернення 12.03.2025).

59.Machine Learning - Polynomial Regression. URL: https://www.w3schools.com/python/python_ml_polynomial_regression.asp (дата звернення 12.03.2025).

60.Gradient Boosting in ML. URL: https://www.geeksforgeeks.org/ml-gradient-boosting/ (дата звернення 12.03.2025).

61.What is Gradient Boosting? URL: https://www.ibm.com/topics/gradient-boosting (дата звернення 12.03.2025).

62.Random Forest Simple Explanation. URL: https://williamkoehrsen.medium.com/random-forest-simple-explanation-377895a60d2d (дата звернення 13.03.2025).

63.What is unsupervised learning? URL: https://cloud.google.com/discover/what-is-unsupervised-learning (дата звернення 13.03.2025).

64.Unsupervised Learning in Machine Learning. URL: https://www.appliedaicourse.com/blog/unsupervised-learning-in-machine-learning/ (дата звернення 13.03.2025).

65.What is k-means clustering? URL: https://www.ibm.com/think/topics/k-means-clustering (дата звернення 13.03.2025).

66.Unsupervised Learning. URL: https://github.com/Falmouth-Games-Academy/comp250-wiki/wiki/Unsupervised-Learning (дата звернення 13.03.2025).

67.What is the Apriori algorithm? URL: https://www.ibm.com/think/topics/apriori-algorithm (дата звернення 13.03.2025).

68.Introduction to Dimensionality Reduction. URL: https://www.geeksforgeeks.org/dimensionality-reduction/ (дата звернення 13.03.2025).

69.Principal Component Analysis (PCA): A Step-by-Step Explanation. URL: https://builtin.com/data-science/step-step-explanation-principal-component-analysis (дата звернення 15.03.2025).

70.What is principal component analysis (PCA)? URL: https://www.ibm.com/think/topics/principal-component-analysis (дата звернення 15.03.2025).

71.Reinforcement Learning. URL: https://www.geeksforgeeks.org/what-is-reinforcement-learning/ (дата звернення 15.03.2025).

72.What is reinforcement learning? URL: https://www.ibm.com/think/topics/reinforcement-learning (дата звернення 15.03.2025).

73.Q-Learning in Reinforcement Learning. URL: https://www.geeksforgeeks.org/q-learning-in-python/ (дата звернення 15.03.2025).

74.SARSA (State-Action-Reward-State-Action) in Reinforcement Learning. URL: https://www.geeksforgeeks.org/sarsa-reinforcement-learning/ (дата звернення 16.03.2025).

75.ML | Monte Carlo Tree Search (MCTS). URL: https://www.geeksforgeeks.org/ml-monte-carlo-tree-search-mcts/ (дата звернення 16.03.2025).

76.What is deep learning? URL: https://www.ibm.com/think/topics/deep-learning (дата звернення 16.03.2025).

77.Mienye I. D., Swart T. G., Obaido G. Recurrent Neural Networks: A Comprehensive Review of Architectures, Variants, and Applications. *Information*. 2024. Vol. 15, no. 9. P. 517.

78.Introduction to Convolution Neural Network. URL: https://www.geeksforgeeks.org/introduction-convolution-neural-network/ (дата звернення 16.03.2025).

79.What is a GNN (graph neural network)? URL: https://www.ibm.com/think/topics/graph-neural-network (дата звернення 16.03.2025).

80.Deep Q-Learning in Reinforcement Learning. URL: https://www.geeksforgeeks.org/deep-q-learning/ (дата звернення 17.03.2025).

81.Deep Q-Learning (DQN). URL: https://medium.com/@samina.amin/deep-q-learning-dqn-71c109586bae (дата звернення 17.03.2025).

82.How DDPG (Deep Deterministic Policy Gradient) Algorithms works in reinforcement learning ? URL: https://medium.com/@amaresh.dm/how-ddpg-deep-deterministic-policy-gradient-algorithms-works-in-reinforcement-learning-117e6a932e68 (дата звернення 17.03.2025).

83.Proximal Policy Optimization (PPO) in Reinforcement Learning. URL: https://www.geeksforgeeks.org/a-brief-introduction-to-proximal-policy-optimization/ (дата звернення 17.03.2025).

84.Asynchronous Advantage Actor Critic (A3C) algorithm. URL: https://www.geeksforgeeks.org/asynchronous-advantage-actor-critic-a3c-algorithm/ (дата звернення 18.03.2025).

85.Stern R. Multi-Agent Path Finding – An Overview. *Artificial Intelligence*. Cham, 2019. P. 96–115.

86.Multi-agent pathfinding with continuous time / A. Andreychuk et al. *Artificial Intelligence*. 2022. Vol. 305. P. 103662.

87.A review of Pareto pruning methods for multi-objective optimization / S. Petchrompo et al. *Computers & Industrial Engineering*. 2022. Vol. 167. P. 108022.

88.Pattanayak S. K., Bhoyar M. Deep Reinforcement Learning for Complex Decision-Making Tasks. *International Journal of Innovative Research in Science,Engineering and Technology*. 2024. Vol. 13, no. 11.

89.Multi-UAV Path Planning for Wireless Data Harvesting With Deep Reinforcement Learning / H. Bayerlein et al. *IEEE Open Journal of the Communications Society*. 2021. Vol. 2. P. 1171–1187.

90.Ikeda T., Shibuya T. Centralized Training with Decentralized Execution Reinforcement Learning for Cooperative Multi-agent Systems with Communication Delay. *2022 61st Annual Conference of the Society of Instrument and Control Engineers (SICE)*, Kumamoto, Japan, 6–9 September 2022. 2022.

91.Erdődi L., Zennaro F. M. The Agent Web Model: modeling web hacking for reinforcement learning. *International Journal of Information Security*. 2021.

92.Guicheng S., Yang W. Review on Dec-POMDP Model for MARL Algorithms. *Smart Communications, Intelligent Algorithms and Interactive Methods*. Singapore, 2022. P. 29–35.

93.Azam M. A., Mittelmann H. D., Ragi S. UAV Formation Shape Control via Decentralized Markov Decision Processes. Algorithms. 2021. Vol. 14, no. 3. P. 91.

94.Stackelberg Actor-Critic: Game-Theoretic Reinforcement Learning Algorithms / L. Zheng et al. *Proceedings of the AAAI Conference on Artificial Intelligence*. 2022. Vol. 36, no. 8. P. 9217–9224.

95.A Multiobjective Optimization Algorithm for Safety and Optimality of 3-D Route Planning in UAV / M. Abdel-Basset et al. IEEE Transactions on Aerospace and Electronic Systems. 2024. P. 1–14.

96.A practical guide to multi-objective reinforcement learning and planning / C. F. Hayes et al. *Autonomous Agents and Multi-Agent Systems*. 2022. Vol. 36, no. 26.

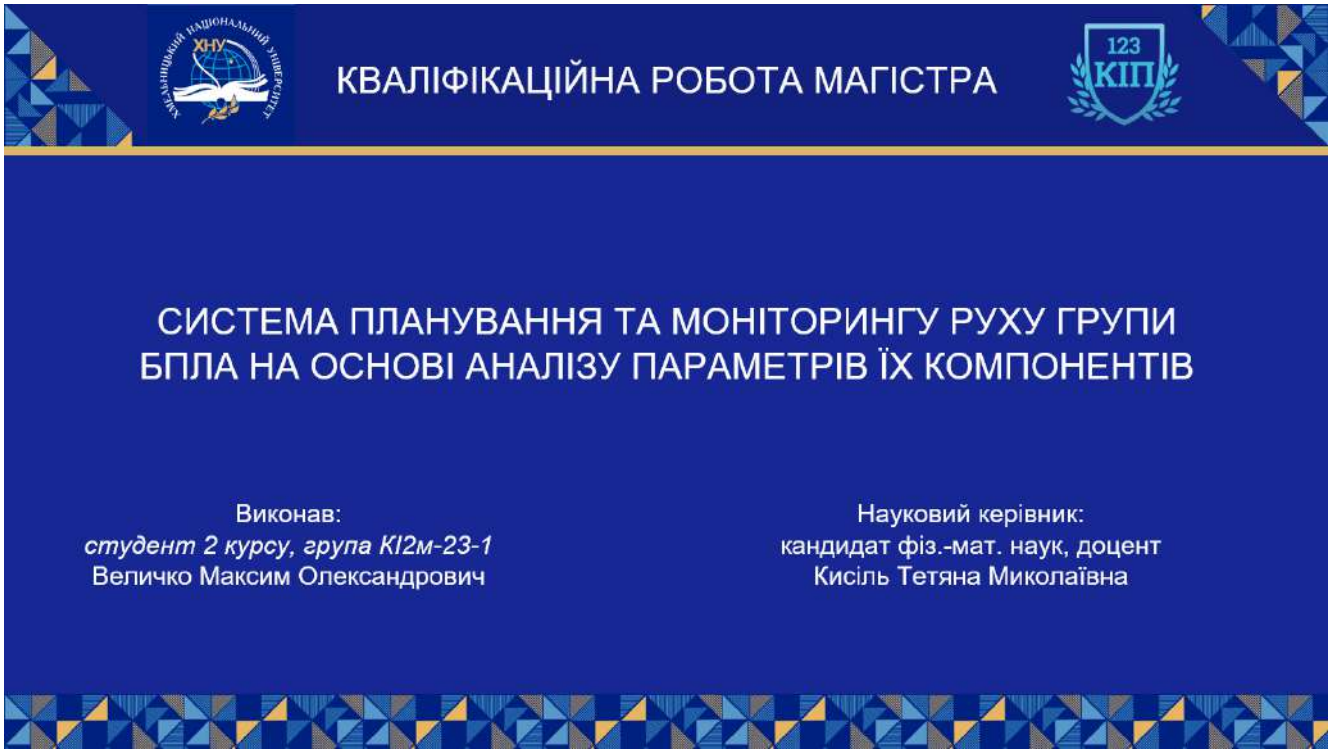
97.Pareto-based bi-objective optimization for robust power allocation in hybrid MIMO phased-array radar system under air defense maneuvering tracking / Z. Li et al. *Expert Systems with Applications*. 2023. P. 121248.

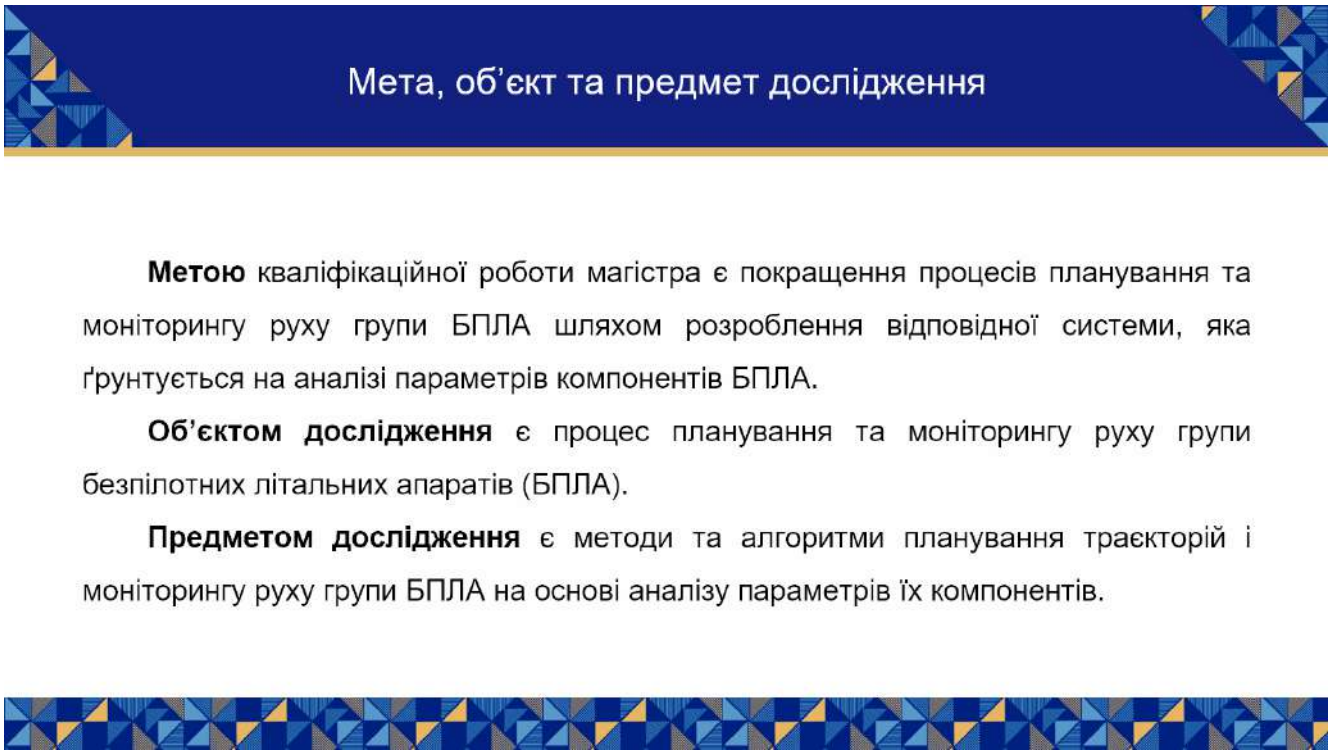
98.Imambi S., Prakash K. B., Kanagachidambaresan G. R. PyTorch. *Programming with TensorFlow*. Cham, 2021. P. 87–104.

99.Velychko M., Kysil T. M. Reinforcement learning method for autonomous flight path planning of multiple uavs. *Computer Systems and Information Technologies*. 2025. №. 2.

ДОДАТОК А
(обов'язковий)

ПРЕЗЕНТАЦІЙНІ МАТЕРІАЛИ





Задачі дослідження

- провести огляд існуючих рішень і методів планування та моніторингу руху групи БПЛА;
- виконати порівняння та вибір алгоритмів для планування та моніторингу руху групи БПЛА;
- покращити обраний метод та алгоритм для планування та моніторингу руху групи БПЛА;
- виконати тестування системи для оцінки ефективності виконання.

Наукова новизна

- удосконалено метод проксимальної оптимізації політики (PPO) для планування та моніторингу руху групи БПЛА в умовах неповної інформації;
- удосконалено структуру нейронної мережі актора та критика алгоритму проксимальної оптимізації політики (PPO) за допомогою шару рекурентної нейронної мережі (RNN), яка забезпечує обробку історичних даних агентів при формуванні політики управління.

Апробація результатів: Velychko M., Kysil T. M. Reinforcement learning method for autonomous flight path planning of multiple uavs. *Computer Systems and Information Technologies*. 2025. №. 2. – **статтю опубліковано у фаховому науковому журналі**

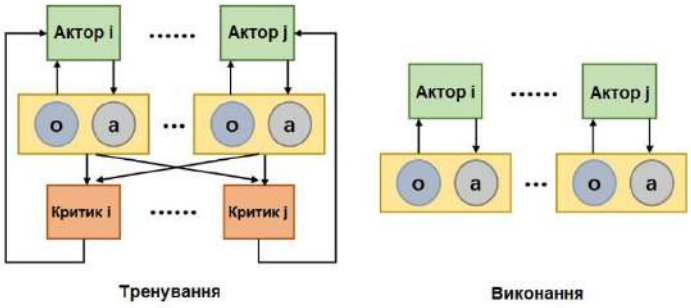
Моделі та методи планування траєкторій руху групи БПЛА в умовах неповної інформації

Автономне планування маршруту для декількох БПЛА вважається повністю кооперативною задачею. Метою всіх учасників такої задачі є отримання максимального командного прибутку. Тому було вирішено створити мультиагентну модель послідовного прийняття рішень в реальному часі за допомогою теоретичної бази децентралізованого частково спостережуваного марковського процесу прийняття рішень (Dec-POMDP). І через складність розв'язку цієї розподіленої моделі для його отримання використано метод навчання з підкріпленням з його алгоритмом акторної критики.

Архітектура методу планування траєкторій руху групи БПЛА в умовах неповної інформації

Оскільки дії кожного агента будуть змінювати середовище, тобто середовище буде нестабільним для кожного агента і призведе до труднощів у навчанні. Тому було обрано архітектуру з централізованим навчанням і децентралізованим виконанням (CTDE).

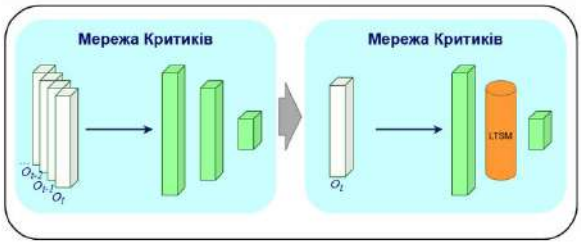
Така архітектура дозволяє кожному агенту спостерігати за діями інших агентів у середовищі, забезпечуючи стабільність середовища.



Проксимальна оптимізація політики

У даному дослідженні для керування БПЛА в рамках політики навчання обрано алгоритм проксимальної оптимізації політики (PPO). Щоб уникнути проблеми цього алгоритма у надмірному оновленні політики між ітераціями було запропоновано нову цільову функцію з обмеженням оновлення політики, яку можна оновлювати невеликими партіями за кілька кроків навчання.

Рекурентна нейронна мережа



Однією з труднощів в задачах автономного планування маршруту з декількома БПЛА є часткове спостереження, що призводить до обмеженої інформації, яку отримують БПЛА.

Тому було додано шар рекурентної нейронної мережі до нейронної мережі критиків в алгоритмі проксимальної оптимізації політики (PPO).

Багатоцільова оптимізація

Спільну функцію винагороди розроблено шляхом декомпозиції загальної задачі на низку підзадач:

$$r_{\text{total}} = \alpha \cdot r_{\text{timecost}} + \beta \cdot r_{\text{security}} + \gamma \cdot r_{\text{coverage}},$$

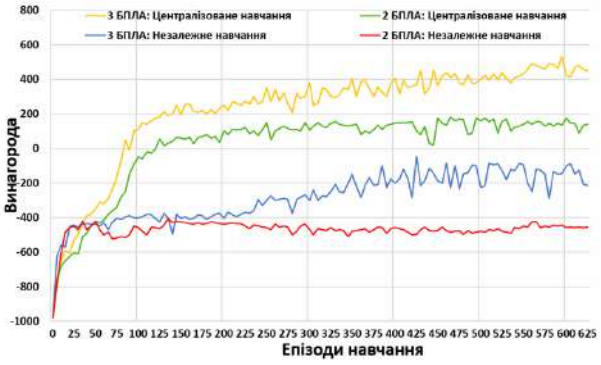
де $r_{\text{security}} = \sum \left|\text{distance}(UAV_i - UAV_j)\right| - \left|\text{distance}(UAV_i - \text{target}_j)\right|$ – це компонент, що сприяє уникненню зіткнень між БПЛА шляхом контролю відстаней між ними, а також стимулює їх наближення до цільових точок;

$r_{\text{coverage}} = \sum \text{new area}_{UAV_i}$ – винагорода за дослідження нових ділянок території;

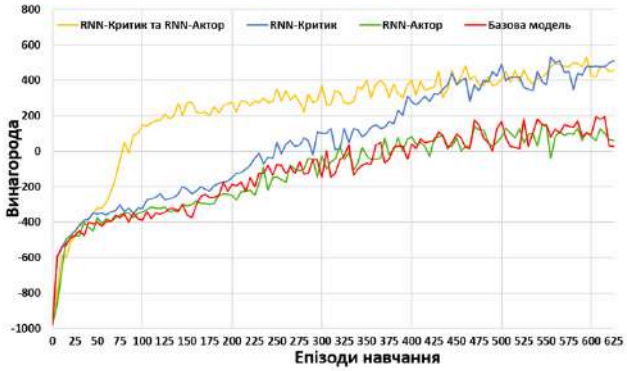
r_{timecost} – компонент, що сприяє мінімізації часу досягнення цільових точок за рахунок скорочення кількості кроків.

Коефіцієнти α, β, γ задаються з урахуванням пріоритетності відповідних критеріїв і можуть змінюватися залежно від вимог до виконання місії.

Результати роботи

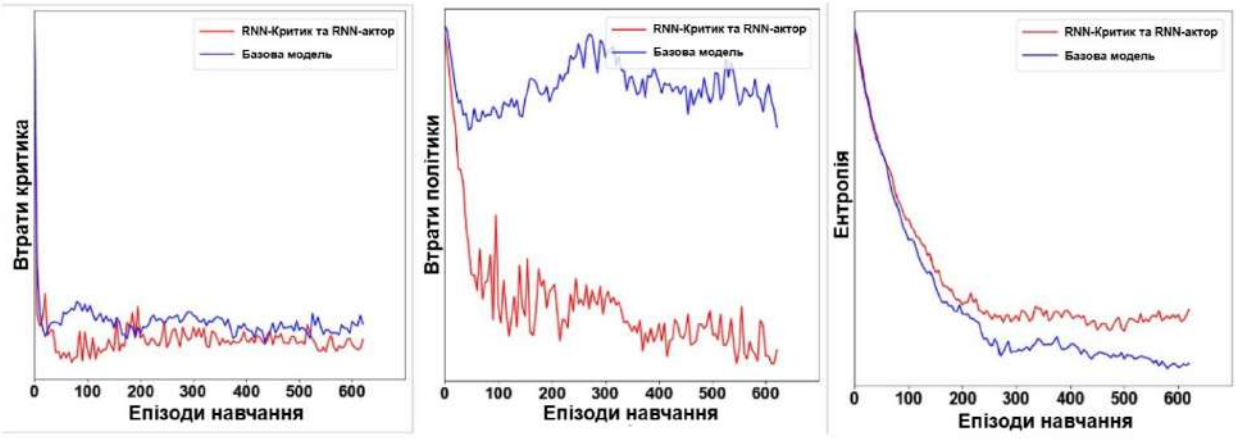


Порівняння CTDE та незалежної архітектури



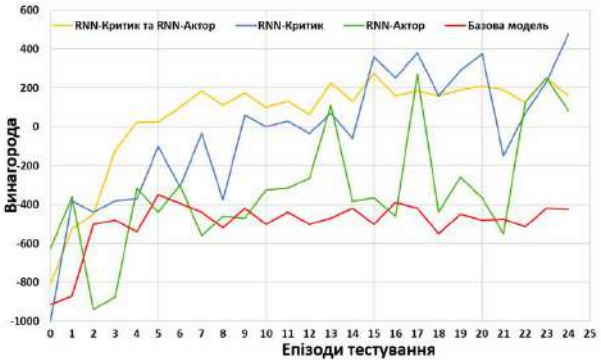
Ефективність моделі після додавання шару RNN до мережі акторів і критиків

Результати роботи

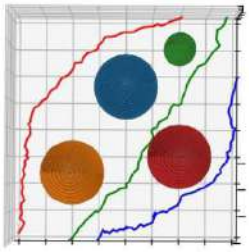


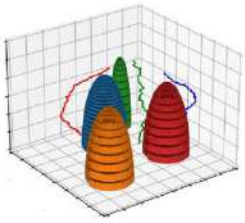
Зміна функцій втрат критика, політики та ентропії протягом тренувальних епізодів

Результати роботи



Порівняння середньої винагороди різних архітектур на етапі тестування





Планування траєкторії руху декількох БПЛА в тривимірному середовищі

Висновки

У роботі за результатами виконаних теоретичних та практичних досліджень розроблено систему планування та моніторингу руху групи БПЛА на основі аналізу параметрів їх компонентів, що враховує умови неповної інформації та динаміку зовнішнього середовища. Реалізовано та вдосконалено алгоритм проксимальної оптимізації політики (PPO), заснований на акторно-критичній архітектурі з використанням CTDE. Для обробки історичних даних та компенсації часткової спостережуваності додано шар рекурентної нейронної мережі (RNN). Також сформульовано функцію багатоцільової винагороди. Проведено тестування її роботи в симульованому середовищі. Розроблений підхід успішно забезпечив побудову маршрутів у режимі реального часу.

Дякую за увагу!

ДОДАТОК Б
(обов'язковий)

НАУКОВА ПРАЦЯ ЗДОБУВАЧА

UDC 629.735.33:004.896

M.O. VELYCHKO[1], T. M. KYSIL[1]
[1] Khmelnytskyi National University

REINFORCEMENT LEARNING METHOD FOR AUTONOMOUS FLIGHT PATH PLANNING OF MULTIPLE UAVS

This study aims to develop a reinforcement learning method for autonomous flight path planning of multiple UAVs under real-world conditions with limited observations and multiple conflicting optimization objectives. The research proposes a multi-agent reinforcement learning approach based on Proximal Policy Optimization (PPO) combined with centralized training and decentralized execution (CTDE). Additionally, a recurrent neural network (RNN) layer is integrated into the critic and actor networks to address partial observability. The reward function is designed to balance time efficiency, safety, and area coverage. Experimental results demonstrate that the proposed method significantly outperforms independent learning approaches in terms of reward accumulation, convergence speed, and decision stability. The CTDE architecture with RNN-enhanced critics proved effective in handling the challenges of multi-agent coordination and partial observability. The trained model enables real-time trajectory planning in three-dimensional environments, surpassing traditional optimization methods. The novelty lies in the application of a multi-agent PPO architecture enhanced by RNNs under CTDE for solving real-time multi-objective optimization problems in UAV path planning. A customized reward structure was developed to simultaneously optimize safety, time, and coverage objectives without retraining. The developed method enables efficient and reliable online trajectory planning for UAV groups, making it applicable in surveillance, search and rescue, and exploration missions where rapid and adaptive decision-making is essential.

Keywords: multiple UAVs, path planning, reinforcement learning , centralized training, decentralized execution, multi-agent systems, PPO algorithm, RNN, CTDE architecture.

М.О. ВЕЛИЧКО[1], Т. М. КИСІЛЬ[1]
[1] Хмельницький національний університет

МЕТОД НАВЧАННЯ З ПІДКРІПЛЕННЯМ ДЛЯ АВТОНОМНОГО ПЛАНУВАННЯ ТРАЄКТОРІЇ ПОЛЬОТУ ДЕКІЛЬКОХ БПЛА

Метою роботи є розробка методу навчання з підкріпленням для автономного планування траєкторій польоту декількох БПЛА в умовах обмеженої видимості середовища та конфліктних цілей оптимізації. Запропоновано багатоагентний підхід навчання з підкріпленням на основі алгоритму проксимальної оптимізації політики (PPO) з використанням архітектури централізованого навчання та децентралізованого виконання (CTDE). Для покращення роботи в умовах часткових спостережень інтегровано рекурентну нейронну мережу в структури акторів і критиків. Розроблено спеціалізовану функцію винагороди, яка враховує показники безпеки, швидкості досягнення цілей та площі покриття території. Результати експериментів показали перевагу запропонованого методу над незалежним навчанням за критеріями швидкості збіжності, стабільності стратегії та величини отриманої винагороди. Структура CTDE із рекурентними мережами дозволила ефективно вирішити проблеми координації між БПЛА та неповної інформації про середовище. Модель забезпечує автономне планування траєкторій у реальному часі у тривимірних середовищах. Наукова новизна полягає в поєднанні методів глибокого навчання з підкріпленням, рекурентних нейронних мереж та архітектури CTDE для вирішення задач багатокритеріальної оптимізації в умовах часткової доступності даних. Розроблений підхід дозволяє підвищити ефективність групової навігації БПЛА, зокрема в сферах розвідки, пошуково-рятувальних операцій і моніторингу, де важливими є автономність, швидкість реагування і надійність.

Ключові слова: планування траєкторій, навчання з підкріпленням, централізоване навчання, децентралізоване виконання, багатоагентні системи, алгоритм PPO, RNN, архітектура CTDE.

Introduction

Nowadays, autonomous flight path planning for multiple UAVs has become a critical challenge due to dynamic environments, limited sensing capabilities, and the need for real-time decision-making. Traditional algorithms struggle with scalability and adaptability in such settings. Reinforcement learning (RL), particularly deep reinforcement learning (DRL), has demonstrated remarkable success in decision-making tasks under uncertainty. However, multi-agent systems introduce additional complexity, including decentralized information and conflicting objectives. This study proposes a reinforcement learning approach based on Proximal Policy Optimization (PPO) and a Centralized Training with Decentralized Execution (CTDE) paradigm. The method addresses the challenges of partial observability by integrating

recurrent neural networks into the actor-critic architecture and formulates a reward function that balances time, safety, and coverage criteria. The goal is to enable autonomous, cooperative, and efficient flight path planning for a group of UAVs operating in real-world conditions.

Related works

Most previous studies on multi-UAV path planning focused on intelligent optimization methods, particularly evolutionary algorithms like Particle Swarm Optimization (PSO) [1–3]. Shao et al. [4] improved PSO for better convergence and obstacle avoidance, while Evan et al. [5] adapted PSO for unknown environments, and Ajeil et al. [6] proposed a hybrid PSO method optimizing path smoothness. However, swarm intelligence algorithms struggle with scalability and real-time performance, making them less suitable for reconnaissance missions.

The rise of Deep Reinforcement Learning (DRL) opened new possibilities for decision-making in dynamic, partially observable environments [7–9]. Reinforcement learning excels where traditional algorithms fail, offering better generalization and real-time inference speed.

Challenges in multi-UAV settings include limited perception, dynamic state changes, and coordination. Fully distributed learning architectures are inefficient for solving Multi-Agent Path Finding (MAPF) problems. Centralized training with decentralized execution (CTDE), first proposed by Lowe [10], addresses these issues by using additional information during training. Jose et al. [11] applied CTDE in vehicle routing, achieving near-optimal solutions, while Marc et al. [12] and Wang et al. [13] demonstrated improvements in UAV conflict resolution and dynamic routing using similar frameworks.

Reward design is critical in reinforcement learning for multi-objective tasks. Simple reward structures risk "reward hacking" [14], while overly complex rewards hurt generalization. The common practice is to transform multi-objective problems into single-objective optimization via weighted reward functions, achieving near-optimality. Li [15] and Xu [16] proposed advanced DRL frameworks to solve multi-objective problems in robotic control and continuous optimization, relevant for UAV path planning where time, safety, and coverage constraints must be considered.

Purpose

Route planning for multiple UAVs can be viewed as a Multi-Agent Path Finding (MAPF) problem, which is a model used to find the optimal path for multiple agents from start positions to destinations without conflicts. In fact, MAPF is a relatively complex optimisation problem with a common goal. The state space of this problem grows exponentially with the number of agents, and it has been proven to be NP-hard. In reconnaissance missions, UAV groups must not only avoid dangerous areas and safely reach target points, but also cover a larger area in a shorter time. However, time and coverage area are in conflict with each other because these are multi-objective optimisation problems, and we must find a trade-off between two or more conflicting objectives to make an optimal decision. It is not possible to find a solution that will achieve optimal fulfilment of all objectives, so for a multi-objective optimisation problem, a set of non-compromised solutions is usually used, which is called a "Pareto solution set".

Proposed technique

Due to the development of deep reinforcement learning (DRL), researchers are actively studying its application for trajectory planning and navigation of UAV groups. Unlike traditional algorithms, DRL performs better in unknown and dynamic environments, providing fast inference and good generalisation in real-time tasks.

This study takes into account partial observations of a group of UAVs, which makes it difficult to make optimal decisions due to the lack of global information. Since the actions of individual UAVs change the state of the environment, incomplete data reduces the effectiveness of learning. To coordinate the actions, a centralised learning architecture with decentralised execution is proposed, which has proven to be effective in multi-agent tasks.

Particular attention is paid to the reward function: too simple a function impairs learning, and too complex reduces generalisation. A multi-factor reward approach with appropriate weighting is optimal. When planning routes for UAVs, time, safety, and coverage are taken into account.

An improved multiagent algorithm based on PPO, a model-free reinforcement learning method that provides adaptability and generalisation, is proposed. The centralised PPO critic network coordinates UAVs through joint observations, and the actor network determines actions. The addition of a recurrent neural network allows historical information to be taken into account to compensate for partial observations. A joint reward function is also developed to learn the optimal policy. After training, each UAV acts based on local data.

During reconnaissance missions, multiple UAVs must plan routes to target points in real time, avoiding collisions and taking into account time and coverage. Autonomous trajectory planning in such conditions is a distributed decision-making problem with partial observations and multi-objective optimisation. Since all UAVs work cooperatively to achieve a common goal, the problem is modelled as a Decentralised Partially Observable Markov Decision Process (Dec-POMDP). Dec-POMDP allows multiple agents to make decisions based on local observations without knowledge of the global state, while being rewarded for a shared long-term benefit. Since the optimal solution of this distributed model has double exponential complexity, reinforcement learning is used to approximate the solution.

In reinforcement learning, the agent optimises its policy by interacting with the environment and receiving rewards. There are methods based on values and policies: the former are difficult to apply in continuous action spaces, and the latter suffer from low efficiency. To solve this problem, actor-critical algorithms have been developed, where an actor generates actions and a critic evaluates their quality, which increases the efficiency of learning. Popular examples: DDPG, PPO, and A3C.

In cooperative multi-agent environments (Dec-POMDP), simple independent learning of agents is inefficient due to the instability of the environment. The MADDPG algorithm solves this problem through centralised critic training, where the input is based on the joint observations of all agents. Actors act on the basis of local information. This architecture of centralised training and decentralised execution (CTDE) ensures the stability and efficiency of multi-agent learning, as in COMA.

In this study, the proximal policy optimisation (PPO) algorithm is chosen for UAV control, which combines the efficiency of the policy gradient with the stability of optimisation due to the constraint of policy changes. PPO is based on an actor-critical architecture and scales well for problems with a continuous action space, making it suitable for real-time UAV trajectory planning.

Typically, policy gradient algorithms have high variability due to excessive policy updates, which leads to learning instability. PPO solves this problem through a special objective function that limits the updates, ensuring that the policy is gradually adapted over several iterations. The algorithm takes into account the difference between the old and new network when updating the parameters. The basis of PPO is the expected reward gradient, which is defined as:

$$\nabla \overline{R}(\tau) = \mathrm{E}_{\tau \cdot \pi\theta(\tau)}\left[A^{\pi}(s_t, a_t) \nabla \log p_{\theta}(a_t \mid s_t) \right], \quad (1)$$

where π_θ – is the policy parameterised by the vector θ;

$A^{\pi}(s_t, a_t)$ – the function of superiority.

The preference function determines how much a particular action is better than the average in a given state and is calculated as:

$$A(s,a) = Q(s,a) - V(s), \quad (2)$$

where $Q^{\pi}(s,a)$ – is the expected total reward after performing the action$a$ a in the state s;

$V^{\pi}(s)$ – expected remuneration from the state of s according to the policy of π.

To ensure the stability of updates, PPO uses the importance sampling technique:

$$\mathbb{E}_{x\sim p}\left[f(x)\right]=\mathbb{E}_{x\sim q}\left[f(x)\frac{p(x)}{q(x)}\right], \tag{3}$$

which allows us to reassess expectations under the new policy using the data collected from the old one.

The main feature of PPO is the use of clipping to stabilise learning. This prevents excessive changes in the policy. The corresponding constrained objective function is as follows:

$$\mathcal{L}^{CLIP}(\theta)=\mathbb{E}_t\left[\min\left(\left(\frac{\pi_\theta(a_t\mid s_t)}{\pi_{\theta_{old}}(a_t\mid s_t)}\right)\hat{A}_t,\operatorname{clip}\left(\frac{\pi_\theta(a_t\mid s_t)}{\pi_{\theta_{old}}(a_t\mid s_t)},1-\epsilon,1+\epsilon\right)\hat{A}_t\right)\right], \tag{4}$$

where $\hat{A}_t$ – is the estimate of the preference function at step t;

ϵ – defines the acceptable limit of policy changes.

This approach strikes a balance between exploring new actions and maintaining an effective strategy, which is especially important for controlling many UAVs in a real-world environment.

Based on the PPO algorithm, the CTDE (Centralised Training with Decentralised Execution) architecture training method was applied and a multi-agent PPO algorithm was developed in a multi-agent environment. Compared to a single-agent environment, the input to the critic network is the joint observation of several UAVs, which is equivalent to the operation of a centralised controller. This approach allows each drone to receive more relevant information for decision-making. The actor network is updated by maximising the following objective function:

$$\mathcal{L}(\theta)=\frac{1}{Bn}\sum_{i=1}^{B}\sum_{k=1}^{n}\left[\min\left(r_{\theta,i}^{k}A_i^k,\operatorname{clip}\left(r_{\theta,i}^{k},1-\epsilon,1+\epsilon\right)A_i^k\right)+\sigma\cdot S_\pi\right], \tag{5}$$

where $r_{\theta,i}^{k}=\dfrac{\pi_\theta(a_i^k\mid o_i^k)}{\pi_{\theta_{\text{old}}}(a_i^k\mid o_i^k)}$ – is the probability ratio between the new and old policies;

A_i^k – the function of superiority;

ϵ – is a restriction parameter;

S_π – the entropy of politics, which facilitates the exploration of new actions.

The critic's network, in turn, is updated by minimising the loss of the value function, which is defined as follows:

$$\mathcal{L}(\phi)=\frac{1}{Bn}\sum_{i=1}^{B}\sum_{k=1}^{n}\max\left[\left(V_\phi(s_i^k)-\mathcal{R}_i\right)^2,\left(\operatorname{clip}\left(V_\phi(s_i^k),V_{\phi_{\text{old}}}(s_i^k)-\varepsilon,V_{\phi_{\text{old}}}(s_i^k)+\varepsilon\right)-\mathcal{R}_i\right)^2\right], \tag{6}$$

where $\mathcal{R}_i^k$ – is a target value (for example, a win or a return estimate);

$V_\phi(s)$ – the state value estimated by the current critic network.

The weights of the two neural networks (actor and critical) are updated in each episode. The update process is shown in figure 1.

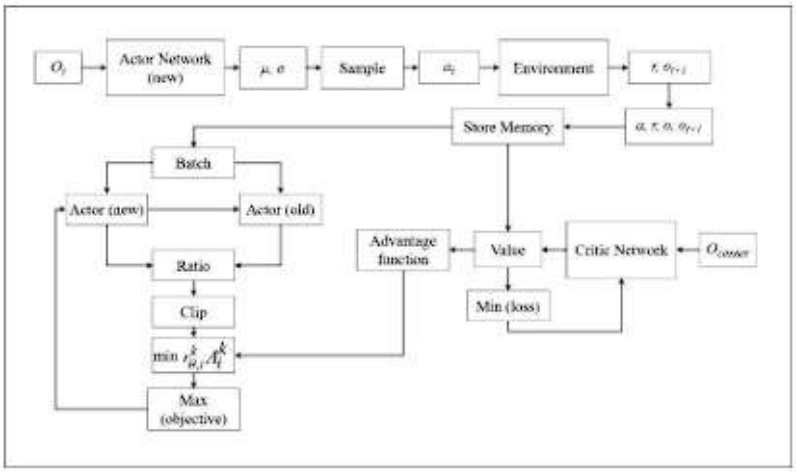


Fig.1. Scheme of updating the weights of the actor-critic network in each episode.

The actor network, receiving the state of the environment o_t as input, generates normal distribution parameters μ, σ, from which the action a_t is selected. After the action is performed, the environment returns the reward r_t and the new state o_{t+1}. The collected data is saved to the memory buffer, after which a data batch is formed and the advantage function is calculated. At the same time, the critical network is updated to minimise losses. Next, the ratio between the old and new policies is calculated, the policy change limitation operation is applied (clipping), and the maximum of the objective function is selected. This helps to stabilise learning and avoid large policy updates.

One of the main problems of autonomous route planning for a group of UAVs is partial observation, due to which agents have limited information. To compensate for this, recurrent neural networks (RNNs) that store historical data are used. The first successful combination of RNNs and reinforcement learning was implemented in the Deep Recurrent Q-Network (DRQN) (figure 2), where one of the linear layers of the DQN was replaced by an RNN, which allowed it to work without stacking frames.

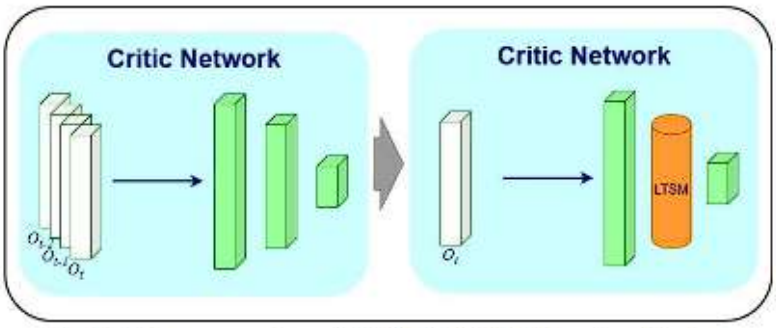


Fig.2. Adding a recurrent layer (LSTM) to the Critical Network architecture.

In this study, an RNN layer is added to the PPO algorithm to process sequences of historical observations. Since classical RNNs have a short-term memory limitation due to gradient decay, the Long Short-Term Memory (LSTM) architecture is used to solve this problem (figure 3). It uses a gate mechanism that allows:

- forgetting unnecessary information (forget gate);
- remember new information (input gate);
- output relevant information to the next step (output gate).

Thanks to this structure, LSTM effectively retains important time dependencies and prevents gradients from damping.

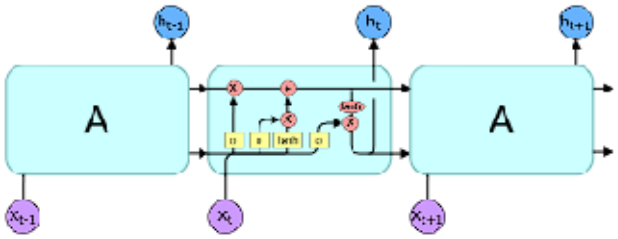


Fig.3. The structure of LSTM: at time t, $X_{(t)}$ is the input signal, C_t is the cell state, h_t is the hidden state.

At each time step t , the input x_t is processed together with the hidden state $h_{(t-1)}$ and the cellular state $C_{(t-1)}$, which allows generating updated values of h_t and C_t.σ

The τ packets are used to update the parameters of the actor and critic networks to maximise $\mathcal{L}(\theta)$ and minimise $\mathcal{L}(\varphi)$ using gradient descent:

$$\tau = [s_t, o_t, a_t, r_t, s_{t+1}, o_{t+1}, a_{t+1}, \ldots] \tag{7}$$

We add a layer to the LSTM network, and two elements $h_{(t,\pi)}$ and $h_{t,V}$ 1 are added to τ, which are changed to:

$$\tau = [s_t, o_t, h_{t,\pi}, h_{t,V}, a_t, r_t, s_{t+1}, o_{t+1}, h_{t+1,\pi}, h_{t+1,V}, a_{t+1}, \ldots], \tag{8}$$

where $h_{t,\pi}$ та $h_{t,V}$ – are the hidden states of the actor network and the critic network, respectively, at time t.

The goal is to maximise the loss function for the actor network:

$$\mathcal{L}(\theta) = \frac{1}{Bn}\sum_{i=1}^{B}\sum_{k=1}^{n}\left[\min\left(r_{\theta,i}^{k}A_i^k, \text{clip}(r_{\theta,i}^{k}, 1-\epsilon, 1+\epsilon)A_i^k\right) + \sigma \cdot S_\pi\right], \tag{9}$$

where $r_{\theta,i}^{k}$ – is the probability ratio between the new and old policies;

A_i^k – the function of superiority;

ϵ – the cutoff value;

S_π – policy that stimulates research.

The study analyses the functional purpose of the network of critics and actors in multi-agent reinforcement learning based on the architecture of centralised training with decentralised execution (CTDE). It is established that the critic network acts as a central controller, processing complete observations, so adding an artificial neural network layer to its structure can significantly improve the efficiency of the model in conditions of partial observation.

The actor network implements the policy of an individual agent, and the addition of an RNN layer has a smaller impact on its performance. Experimental results confirmed the feasibility of using the CTDE architecture to solve this problem.

Autonomous route planning for a group of UAVs is considered as a multi-objective optimisation problem that takes into account time, coverage area, and system safety. Optimisation involves the selection of decision variables in a discrete space.

This approach is similar in nature to the "action selection" task in reinforcement learning. The combination of "offline learning" and "online decision-making" within the framework of deep reinforcement learning allows for real-time implementation of multi-objective optimisation. Thus, deep reinforcement learning methods are an effective tool for solving multi-objective optimisation problems in systems with several UAVs, providing high generalisability of the built models.

The reward function was developed based on the principles of multi-objective optimisation and previous knowledge in the field of navigation by decomposing the overall task into a number of sub-tasks. The joint reward function was formed taking into account the constraints on safety, execution time and territory coverage. The reward function has the following form:

$$r_{\text{total}} = \alpha \cdot r_{\text{timecost}} + \beta \cdot r_{\text{security}} + \gamma \cdot r_{\text{coverage}}, \tag{10}$$

where $r_{\text{security}} = \sum \left| \text{distance}(UAV_i - UAV_j) \right| - \left| \text{distance}(UAV_i - \text{target}_j) \right|$ – is a component that helps to avoid collisions between UAVs by controlling the distances between them, and encourages them to approach target points;

$r_{\text{coverage}} = \sum \text{new area}_{UAV_i}$ – reward for exploring new areas of the territory, which encourages UAVs not only to achieve goals, but also to actively explore the environment;

r_{timecost} – a component that helps to minimise the time to reach target points by reducing the number of steps.

The coefficients α, β, γ are set taking into account the priority of the relevant criteria and can be changed depending on the mission requirements. Thanks to this structure of the reward function, the system can adapt to different scenarios while maintaining an optimal balance between efficiency, safety and coverage. This approach makes it possible to turn a multi-criteria problem into a single-criteria problem with the possibility of further adjusting the aggressiveness of the agent's strategy without complete retraining. A multi-objective optimisation problem can be solved by means of multi-objective reinforcement learning.

To solve the problem of multicriteria policy optimisation, a gradient multicriteria approach is used. This approach is aimed at maximising the weighted sum of rewards, where the weight vector ω determines the importance of each of the objective functions. This leads to a change in the appearance of the policy gradient.

The functional of the objective of a multi-criteria policy can be represented as follows:

$$\mathcal{J}(\theta, \omega) = \omega^{\top} F(\pi) = \sum_{i=1}^{m} \omega_i f_i(\pi) = \sum_{i=1}^{m} \omega_i \mathcal{J}_i^{\pi}, \tag{11}$$

where θ – is the policy settings;

$\omega \in \mathbb{R}^m$ – a vector of weights for each target;

$\mathcal{J}_i^{\pi}$ – goal functionality for the i task.

The gradient of this function by policy parameters is as follows:

$$\nabla_\theta \mathcal{J}(\theta,\omega) = \sum_{i=1}^{m} \omega_i \nabla_\theta \mathcal{J}_i(\theta) \tag{12}$$

Which can then be represented as a mathematical expectation:

$$\begin{aligned} &= \mathbb{E}\left[\sum_{t=0}^{T} \omega^{\top} A^{\pi}(s_t,a_t)\nabla_\theta \pi_\theta(a_t \mid s_t)\right] \\ &= \mathbb{E}\left[\sum_{t=0}^{T} A_\omega^{\pi}(s_t,a_t)\nabla_\theta \pi_\theta(a_t \mid s_t)\right], \end{aligned} \tag{13}$$

where A^{π} – is an advantage function;

A_ω^{π} – weighted preference, which takes into account all goals with appropriate weights.

Experiments

In this study, the PyTorch library was used to build a three-level actor neural network and critique the PPO algorithm. Centralised training with decentralised execution (CTDE) was implemented to coordinate the UAV's actions. The key difference between centralised and independent training is the composition of the input data for the critic network.

In the experiment, the local observations of all UAVs were aggregated into a single high-dimensional vector, which formed a joint observation for the critic's network (O_{centre}). The actor network received individual observations O_i of each UAV (see figure 4).

Compared to independent training, four control experiments were conducted. The network architecture included three layers: the first and third layers were fully connected, and the second layer was a recurrent LSTM layer. To verify the feasibility of LSTM, a variant of the architecture without the recurrent layer was created, which allowed for a comparative analysis and confirmation of the advantages of the CTDE architecture.

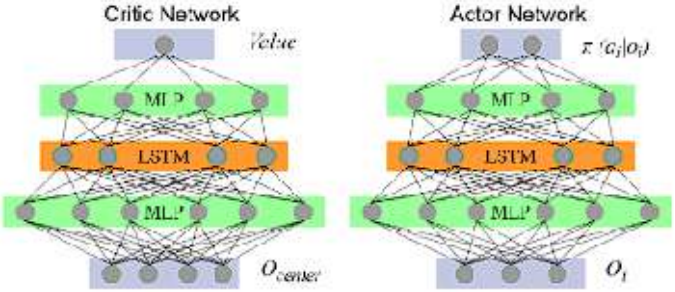


Fig.4. The network of actors and the network of critics, as well as all the agents that share these networks.

The proximal policy optimisation produces random policies, which means that the outputs of the actor network are μ, σ, which are the mathematical expectation and variance of a Gaussian distribution, and the output action is a random sample from this Gaussian distribution.

In the process of training agents using the PPO method, we used the fixed hyperparameters shown in table 1.

Table 1

Neural network parameters and PPO agent training

№	Parameter	Meaning
1	Episode	625
2	Episode length	200
3	Rollout thread	16
4	Clip	0,2
5	Discount	0,99
6	Entropy coefficient	0,1
7	Buffer size	500
8	Batch size	32
9	FC layer dim	128
10	RNN hidden dim	64
11	Activation	ReLU
12	Optimizer	Adam

After performing one million training steps and analysing the experimental results, it was found that the PPO algorithm implemented on the basis of centralised training with decentralised execution demonstrated higher efficiency compared to independent training in the task of autonomous trajectory planning for a group of UAVs.

As can be seen from the results shown in figure 5, fully independent and distributed learning methods have difficulty achieving adequate performance in multi-UAV tasks. The introduction of the CTDE architecture significantly improved the efficiency: the value of the reward function became positive, and the performance increased even as the number of UAVs increased.

This confirms the feasibility of using CTDE to solve distributed tasks. The central controller coordinates the actions of all UAVs, but with more than six UAVs, the efficiency decreases due to the increase in the size of the joint observation space.

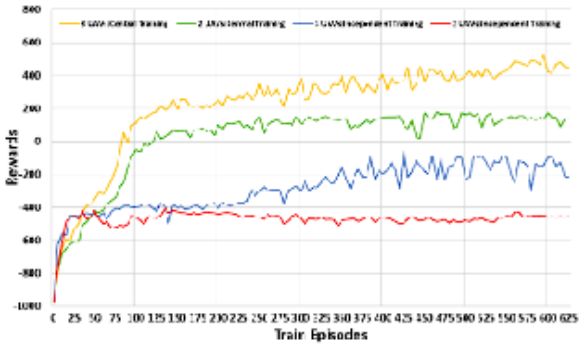


Fig.5. Comparison of CTDE and independent architecture.

Experiments were conducted to test whether the addition of a recurrent neural network (RNN) layer can solve the problem of training many UAVs in the face of incomplete information.

The experimental results (figure 6) showed that adding an RNN layer to the actor and critic networks significantly improves the model's performance. Including the RNN layer in the critics' network provided an improvement, but the convergence process was slower. Adding an RNN layer to the actor network alone did not lead to a significant performance gain and did not solve the problem of partial observations when using multiple UAVs.

These results confirm that in the CTDE architecture, the critic network functions as a central controller, coordinating the UAV's actions, and adding an RNN layer to this network helps to compensate for the problem of incomplete information.

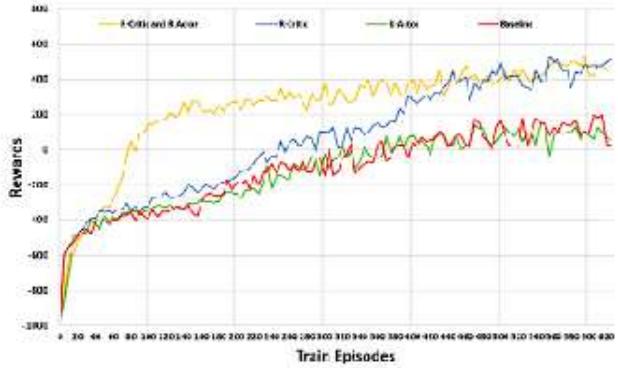


Fig.6. Model performance after adding an RNN layer to the network of actors and critics.

Reinforcement learning demonstrates significantly faster execution speeds than traditional swarm intelligence algorithms, making it suitable for solving real-time problems (figure 8). After training, the parameters of the neural network were fixed, ensuring the stability of the model's behaviour during execution. Solving the navigation task with this model showed higher and more stable rewards compared to other approaches.

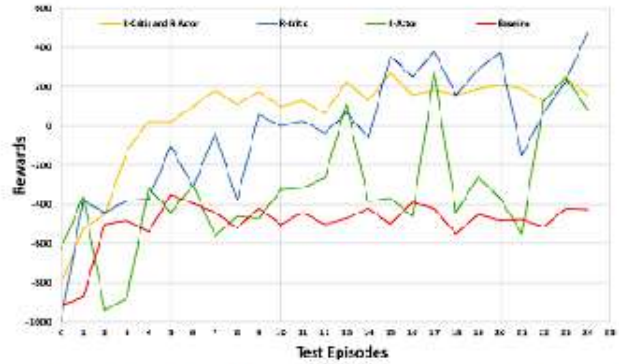


Fig.8. Comparison of the average reward of different architectures at the testing stage.

Conclusions

The study successfully demonstrates that reinforcement learning methods, specifically PPO with CTDE architecture and recurrent neural enhancements, significantly improve the efficiency and reliability of multi-UAV autonomous flight path planning. The centralized critic provides comprehensive coordination based on collective observations, while decentralized actors ensure flexible execution. The developed reward structure effectively balances multiple mission objectives, leading to better overall system performance. Experimental results confirm that the proposed approach outperforms traditional and independent learning methods, highlighting its potential for deployment in complex, dynamic environments requiring real-time decision-making and collaboration among multiple agents.

References

1. Tang J., Duan H., Lao S. Swarm intelligence algorithms for multiple unmanned aerial vehicles collaboration: a comprehensive review. *Artificial Intelligence Review*. 2022.

2. Reconnaissance Mission Conducted by UAV Swarms Based on Distributed PSO Path Planning Algorithms / Y. Wang et al. *IEEE Access*. 2019. Vol. 7. P. 105086–105099.

3. An Accurate UAV 3-D Path Planning Method for Disaster Emergency Response Based on an Improved Multiobjective Swarm Intelligence Algorithm / Y. Wan et al. *IEEE Transactions on Cybernetics*. 2022. P. 1–14.

4. Efficient path planning for UAV formation via comprehensively improved particle swarm optimization / S. Shao et al. *ISA Transactions*. 2020. Vol. 97. P. 415–430.

5. Collision-Free Autonomous Robot Navigation in Unknown Environments Utilizing PSO for Path Planning / E. Krell et al. *Journal of Artificial Intelligence and Soft Computing Research*. 2019. Vol. 9, no. 4. P. 267–282.

6. Multi-objective path planning of an autonomous mobile robot using hybrid PSO-MFB optimization algorithm / F. H. Ajeil et al. *Applied Soft Computing*. 2020. Vol. 89. P. 106076.

7. Deep-Sarsa Based Multi-UAV Path Planning and Obstacle Avoidance in a Dynamic Environment / W. Luo et al. *Lecture Notes in Computer Science*. Cham, 2018. P. 102–111.

8. Distributed Energy-Efficient Multi-UAV Navigation for Long-Term Communication Coverage by Deep Reinforcement Learning / C. H. Liu et al. *IEEE Transactions on Mobile Computing*. 2020. Vol. 19, no. 6. P. 1274–1285.

9. Multi-UAV Path Planning for Wireless Data Harvesting With Deep Reinforcement Learning / H. Bayerlein et al. *IEEE Open Journal of the Communications Society*. 2021. Vol. 2. P. 1171–1187.

10. Lowe, R.; Wu, Y.I.; Tamar, A.; Harb, J.; Pieter Abbeel, O.; Mordatch, I. Multi-agent actor-critic for mixed cooperative-competitive environments. In Proceedings of the Advances in Neural Information Processing Systems. Long Beach. CA. USA. 4–9 December 2017. Volume 30.

11. Vera J. M., Abad A. G. Deep Reinforcement Learning for Routing a Heterogeneous Fleet of Vehicles. *2019 IEEE Latin American Conference on Computational Intelligence (LA-CCI)*, Guayaquil, Ecuador, 11–15 November 2019. 2019.

12. Brittain M., Wei P. Autonomous Separation Assurance in An High-Density En Route Sector: A Deep Multi-Agent Reinforcement Learning Approach. *2019 IEEE Intelligent Transportation Systems Conference - ITSC*, Auckland, New Zealand, 27–30 October 2019. 2019.

13. Cooperative Reinforcement Learning Aided Dynamic Routing in UAV Swarm Networks / Z. Wang et al. *ICC 2022 - IEEE International Conference on Communications*, Seoul, Korea, Republic of, 16–20 May 2022. 2022.

14. Ibarz, B.; Leike, J.; Pohlen, T.; Irving, G.; Legg, S.; Amodei, D. Reward learning from human preferences and demonstrations in atari. In Proceedings of the Advances in Neural Information Processing Systems. Montreal, Canada. 3–8 December 2018. Volume 31.

15. Li K., Zhang T., Wang R. Deep Reinforcement Learning for Multiobjective Optimization. *IEEE Transactions on Cybernetics*. 2020. P. 1–12.

16. Xu, J.; Tian, Y.; Ma, P.; Rus, D.; Sueda, S.; Matusik, W. Prediction-guided multi-objective reinforcement learning for continuous robot control. In Proceedings of the International Conference on Machine Learning. Virtual. 13–18 July 2020. P. 10607–10616.

Maksym Velychko Максим Величко	Master Student of Computer Engineering & Information Systems Department, Khmelnytskyi National University E-mail: vmaks230@gmail.com	Магістрант кафедри комп'ютерної інженерії та інформаційних систем, Хмельницький національний університет
Tetiana Kysil Тетяна Кисіль	Candidate of Physical and Mathematical Sciences, Associate Professor of Computer Engineering & Information Systems Department, Khmenlnytskyi National University ORCID: 0000-0002-4094-3500 E-mail: kysil_tanya@ukr.net	Кандидат фізико-математичних наук, доцент кафедри комп'ютерної інженерії та інформаційних систем, Хмельницький національний університет

Anti-Plagiarism v-15.274 Educational

The maximum coincidence with one document 19.0%

Dictionaries check: en_US, ru_RU, ua_UA. **Errors in the documents: 12%**

	Document		Sum coincidence on the DB	
ID: 241571 Title: МКР Система планування та моніторингу руху групи БПЛА на основі аналізу параметрів їх компонентів Added in a DB: 2025-05-21 Authors: Максим ВЕЛИЧКО Heads: Тетяна КИСІЛЬ Consultants: Opponents:	Symbols	Lexemes	Symbols	Lexemes
	129002	993	25017 (19%)	197 (20%)

Plagiarism sources

ID	Description	Plagiarism presence in the document	
		Symbols	Lexemes
193939	Title: Звіт з ПДП Система планування та моніторингу руху групи БПЛА на основі аналізу параметрів їх компонентів Added in a DB: 2025-03-24 Authors: Величко М.О. Heads: Кисіль Т.М. Consultants: Opponents:	23889 (19.0%)	175 (18.0%)

Протокол аналізу звіту подібності експертом

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Максим ВЕЛИЧКО

Співавтор:

Назва: Величко_Система планування та моніторингу руху групи БПЛА на основі аналізу параметрів їх компонентів

Експерт:

Підрозділ: Кафедра комп'ютерної інженерії та інформаційних систем

Коефіцієнт подібності 1:3.4%

Коефіцієнт подібності 2:0.3%

Мікропробіли: 260

Заміна букв: 1

Інтервали: 0

Білі знаки: 1

Дата створення звіту: 2025-05-21 11:24:09.0

Після аналізу Звіту подібності констатую наступне:

☒ **Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.**

☐ **Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.**

☐ **Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедурам. Таким чином робота не приймається.**

Обгрунтування:

2025-05-21 Доцент Андрій Нічепорук

Дата експерт

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Величко Максим Олександрович

Тема: Система планування та моніторингу руху групи БПЛА на основі аналізу параметрів їх компонентів

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість сторінок записки 84

1. Короткий зміст роботи та прийнятих рішень: Метою кваліфікаційної роботи є покращення процесів планування та моніторингу руху групи БПЛА шляхом розроблення відповідної системи, яка ґрунтується на аналізі параметрів компонентів БПЛА для забезпечення оптимального виконання польотних завдань та підвищення ефективності роботи групи.

2. Висновок про відповідність роботи дипломному завданню: Кваліфікаційна робота магістра відповідає виданому завданню

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У першому розділі проведено огляд предметної області, досліджено відомі підходи та виклики в цій сфері. У другому розділі проаналізовано алгоритми машинного навчання для планування та моніторингу маршрутів руху групи БПЛА. У третьому розділі запропоновано метод планування та моніторингу маршрутів руху групи БПЛА. У четвертому розділі описано архітектуру розробленої системи та проведено тестування її роботи в симульованому середовищі.

4. Позитивні сторони роботи: Запропонована система підвищила ефективність роботи групи БПЛА завдяки впровадженню покращеного методу планування та моніторингу руху. Система здатна автоматично планувати маршрути, координувати роботу групи БПЛА, здійснювати моніторинг стану компонентів та адаптуватися до змін у реальному часі та може бути інтегрована в існуючі платформи для управління БПЛА, забезпечуючи автоматизований контроль та оптимізацію траєкторій.

5. Негативні сторони роботи:

6. Оцінка графічного оформлення та пояснювальної записки роботи: Пояснювальна записка оформлена коректно, згідно з діючими стандартами оформлення документації.

7. Відгук про роботу в цілому: Робота виконана на високому науково-технічному рівні.

8. Інші зауваження: _____

9. Оцінка дипломної роботи: відмінно/A (4.75).

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) _____

Мартинюк Валерій Володимирович,
зав. каф. АКІТ та Р, ХНУ

"22" 05 2025р. _____(підпис)

Завідувачу кафедри КІІС
доктору філософії, доценту
Ользі ПАВЛОВІЙ

Величка Максима Олександровича
ПІБ здобувача вищої освіти

ФІТ, 2 курсу, групи КІ2м-23-1

ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності у Хмельницькому національному університеті» від 01.07.2022, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений (а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений(а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (StrikePlagiarism та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

29 квітня 2025 року

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ

КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Система планування та моніторингу руху групи БПЛА на основі аналізу параметрів їх компонентів

Автор: Величко Максим Олександрович

Спеціальність: 123 – Комп'ютерна інженерія

Освітня програма: освітньо-наукова

Науковий керівник: Кисіль Тетяна Миколаївна, к.ф-м.н., доцент

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

1) запозичення розміщені в розділах є збіг зі звітом з науково-дослідної практики автора Максима Величка "Система планування та моніторингу руху групи БПЛА на основі аналізу параметрів їх компонентів", який було додано в репозитраій ХНУ 24 березня 2025 року;
2) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
3) окремі виявлені збіги є загальновживаними фразами або виразами, про що свідчить посилання системи на збіг з джерелами на один фрагмент речення;
4) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів зі україномовними скороченнями індексів в формулах, що не є модифікацією тексту.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості StrikePlagiarism, складає 3.36 % і адресується до 37 першоджерела; та системою Anti-Plagiarism складає 19%, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи _____ Тетяна КИСІЛЬ

Гарант ОП _____ Олег САВЕНКО

Завідувач кафедри КПС _____ Ольга ПАВЛОВА