

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління
Кафедра Комп'ютерних інтелектуальних технологій та систем

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

рівень вищої освіти другий (магістерський)

Інтелектуальна комп'ютерна система планування завдань в командній
роботі з БПЛА

Виконав:

здобувач 2 року навчання,

групи КІТм-23-1

Антон КІРЄЄВ

Спеціальність 123 Комп'ютерна інженерія

Тип програми освітньо-професійна

Освітня програма Комп'ютерні

інтелектуальні технології

Керівник проф. Наталія АКСАК

Допускається до захисту

Зав. кафедри

Олег РУДЕНКО

(підпис)

2024 р.

Харківський національний університет радіоелектроніки

Факультет	Комп'ютерної інженерії та управління
Кафедра	Комп'ютерних інтелектуальних технологій та систем
Рівень вищої освіти	другий (магістерський)
Спеціальність	123 Комп'ютерна інженерія
Тип програми	освітньо-професійна
Освітня програма	Комп'ютерні інтелектуальні технології

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

« _____ » _____ 202_ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві _____ Кіресву Антону Юрійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи _____ Інтелектуальна комп'ютерна система планування завдань в командній роботі з БПЛА

затверджена наказом по університету від " 28 " жовтня 2024 р. № 1156Ст

2. Термін подання студентом роботи до екзаменаційної комісії _____ 13.01.2025

3. Вхідні дані до роботи _____

Алгоритми навчання (PyTorch/TensorFlow)

Динамічне планування

Коллективне прийняття рішень

Агентна система (JADE)

Середовище моделювання GAMA

4. Перелік питань, що потрібно опрацювати в роботі _____

Аналіз предметної області та постановка задачі

Методи планування та управління завданнями

Модель командної роботи людини і БПЛА

Планування завдань на базі агентів

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням кафедри)

10 слайдів

6. Консультанти розділів роботи и (п.6 включається до завдання за наявністю консультантів згідно до наказу, зазначеному у п.1)

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

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Огляд і аналіз сучасного стану поставленої задачі	28.10.24 - 10.11.24	Виконано
2	Постановка задачі кваліфікаційної роботи	11.11.24 - 13.11.24	Виконано
3	Обґрунтування мети вирішення поставленої задачі	14.11.24 - 20.11.24	Виконано
4	Вибір та розробка моделі командної роботи	21.11.24 - 18.12.24	Виконано
5	Оформлення матеріалів кваліфікаційної роботи	19.12.24 - 08.01.25	Виконано
6	Розробка презентації	09.01.25 - 13.01.25	Виконано
7	Подання кваліфікаційної роботи на рецензування	16.01.2025	Виконано
8	Попередній захист	21.01.2025	Виконано
9	Захист роботи	23.01.2024	Виконано

Дата видачі завдання 28 жовтня 2024 р.

Здобувач _____
(підпис)

Керівник роботи _____
(підпис)

Проф Наталія АКСАК
(посада, ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 96 с., 9 рис., 2 дод., 33 джерела.

АГЕНТНА КООРДИНАЦІЯ, НЕБЕЗПЕЧНЕ СЕРЕДОВИЩЕ, РОЗПОДІЛ РЕСУРСІВ, ДИНАМІЧНЕ ПЛАНУВАННЯ, БПЛА

Метою кваліфікаційної роботи є розробка інтелектуальної комп'ютерної системи планування завдань у командній роботі з безпілотними літальними апаратами (БПЛА), яка включає алгоритми оптимізації завдань, координацію дій та інтеграцію даних з різних джерел для забезпечення ефективності виконання місій.

У кваліфікаційній роботі запропонована архітектура поєднує симуляцію, навчання, управління завданнями та інтеграцію апаратних засобів, що забезпечує ефективне рішення для командної роботи БПЛА, розроблена система командної роботи безпілотних літальних апаратів (БПЛА), в якій інтегровано різні модулі, такі як коактивне проектування, планування завдань на основі агентів, глибоке навчання з підкріпленням та вибір дій зі змішаною ініціативою.

Об'єктом дослідження є інтелектуальні комп'ютерні системи, зокрема система планування завдань в командній роботі з безпілотними літальними апаратами (БПЛА). Предмет дослідження – методи і технології планування завдань у командній роботі, зокрема метод коактивного проектування взаємозалежностей між учасниками команди. А також агентний планувальник завдань, який автоматично декомпозує складні завдання для учасників команди, забезпечуючи ефективну координацію і виконання завдань в умовах взаємозалежності.

Матеріали, розроблені в рамках цієї роботи, можуть бути впроваджені в промисловість для поліпшення координації команд БПЛА в різних сферах, таких як сільське господарство, моніторинг довкілля та військові операції.

ABSTRACT

Master's thesis:: 96 pages, 9 figures, 2 appendices, 33 sources.

AGENT COORDINATION, HAZARDOUS ENVIRONMENT, RESOURCE ALLOCATION, DYNAMIC PLANNING, UAV

The purpose of the qualification work is to develop an intelligent computer system for task planning in teamwork with unmanned aerial vehicles (UAVs), which includes task optimization algorithms, coordination of actions and integration of data from various sources to ensure the effectiveness of mission performance.

The qualification work proposes an architecture based on simulation, training, task management and hardware integration that provides an effective solution for BPLA teamwork, and a system for unmanned aerial vehicles (BPLA) teamwork is developed, in which various modules are integrated, such as coactive design, agent-based task planning, deep reinforcement learning and mixed-initiative action selection.

The object of the research is intelligent computer systems, in particular, a task planning system in teamwork with unmanned aerial vehicles (UAVs). The subject of the research is methods and technologies for task planning in teamwork, in particular, a method for coactive design of interdependencies between team members. As well as an agent task planner that automatically decomposes complex tasks for team members, ensuring effective coordination and task execution in conditions of interdependence.

The materials developed within the framework of this work can be implemented in industry to improve the coordination of UAV teams in various areas, such as agriculture, environmental monitoring, and military operations.

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет

Комп'ютерної інженерії та управління

Кафедра

Комп'ютерних інтелектуальних технологій та систем

АНОТАЦІЯ **КВАЛІФІКАЦІЙНОЇ РОБОТИ**

рівень вищої освіти другий (магістерський)

Інтелектуальна комп'ютерна система планування завдань в командній
роботі з БПЛА

Виконав:

здобувач II курсу, групи КІТм-23-1

Антон КІРЄЄВ

Спеціальність 123 Комп'ютерна інженерія

Тип програми освітньо-професійна

Освітня програма Комп'ютерні інтелектуальні
технології

Керівник проф. Наталія АКСАК

2024 р.

АНОТАЦІЯ

Кіреєв А.Ю. Інтелектуальна комп'ютерна система планування завдань в командній роботі з БПЛА. – Магістерська кваліфікаційна робота.

Актуальність роботи. Безпілотні літальні апарати (БПЛА) корисні для виконання небезпечних і динамічних завдань, таких як пошуково-рятувальні роботи, спостереження за лісом та антитерористичні операції. Ці завдання можуть бути вирішені краще завдяки співпраці декількох БПЛА під наглядом людини. Однак людині все ще важко відстежувати, розуміти, прогнозувати і контролювати поведінку БПЛА через складність завдань, а також через алгоритми машинного навчання і планування, що використовуються в «чорному ящику». Безпілотні літальні апарати (БПЛА) та мультиагентні системи (МАС) знаходяться на передовій сучасних технологій, які дозволяють виконувати різноманітні завдання в автоматизованому режимі. БПЛА, часто відомі як дрони, використовуються для виконання широкого спектру завдань, від військових операцій до комерційних застосувань, таких як доставка товарів, моніторинг сільськогосподарських угідь, та зйомка. Їхня здатність виконувати місії без прямого втручання людини дозволяє знижувати ризики для операторів та збільшувати ефективність операцій.

Мультиагентні системи, з іншого боку, включають в себе координацію багатьох автономних агентів, які можуть працювати спільно для досягнення певних цілей. Такі системи знаходять застосування у різних сферах, включаючи управління трафіком, розподіл ресурсів, та рятувальні операції. Вони дозволяють розподіляти складні завдання між агентами, що може підвищити ефективність та гнучкість системи в цілому.

Метою роботи є розробка інтелектуальної комп'ютерної системи планування завдань у командній роботі з безпілотними літальними апаратами (БПЛА), яка включає алгоритми оптимізації завдань, координацію дій та інтеграцію даних з різних джерел для забезпечення ефективності виконання місій.

Об'єктом дослідження магістерської кваліфікаційної роботи є інтелектуальні комп'ютерні системи, зокрема система планування завдань в командній роботі з безпілотними літальними апаратами (БПЛА). Це система, що забезпечує ефективну організацію та координацію завдань між учасниками команди (включаючи БПЛА та операторів) для досягнення поставлених цілей.

Предмет дослідження – методи і технології планування завдань у командній роботі, зокрема метод коактивного проектування взаємозалежностей між учасниками команди. А також агентний планувальник завдань, який автоматично декомпозує складні завдання для учасників команди, забезпечуючи ефективну координацію і виконання завдань в умовах взаємозалежності.

Методи дослідження. Для визначення теоретичних аспектів та основних підходів до планування завдань в командній роботі з БПЛА, використовується метод літературного аналізу. Огляд сучасних досліджень та публікацій дозволяє сформулювати загальне уявлення про наявні методи, алгоритми та технології, що використовуються в даній галузі. Використання методів комп'ютерного моделювання для створення симуляцій планування завдань в командній роботі з БПЛА. Це дозволяє отримати експериментальні дані, що демонструють ефективність різних підходів до планування завдань, взаємодії учасників та автоматичної декомпозиції завдань. Для цього використовуються спеціалізовані програмні продукти або створені власні моделі, які забезпечують тестування алгоритмів у контрольованих умовах. Проведення експериментів з реальними або віртуальними системами БПЛА, що дозволяє оцінити роботу запропонованого агентного планувальника завдань та методу коактивного проектування взаємозалежностей. Це експерименти з виконанням певних завдань у реальних або смодельованих умовах, з оцінкою часу виконання завдань, точності планування та ефективності взаємодії учасників команди. Зібрані під час експериментів дані аналізуються за допомогою статистичних методів, що дозволяє зробити висновки про ефективність запропонованих методів. Для цього використовуються статистичні інструменти для оцінки точності планування, порівняння результатів різних підходів та виявлення закономірностей. Оскільки дослідження пов'язане з агентними системами,

використовується метод моделювання агентів для розробки та тестування агентного планувальника завдань. Цей метод дозволяє реалізувати модель взаємодії агентів у команді (включаючи БПЛА), розподілу завдань і автоматичної декомпозиції складних завдань на підзавдання.

Наукова новизна. Запропонована архітектура поєднує симуляцію, навчання, управління завданнями та інтеграцію апаратних засобів, що забезпечує ефективне рішення для командної роботи БПЛА, розроблена система командної роботи безпілотних літальних апаратів (БПЛА), в якій інтегровано різні модулі, такі як коактивне проектування, планування завдань на основі агентів, глибоке навчання з підкріпленням та вибір дій зі змішаною ініціативою.

Практична цінність отриманих результатів. Розроблена методологія може бути впроваджена в промисловість для поліпшення координації команд БПЛА в різних сферах, таких як сільське господарство, моніторинг довкілля та військові операції.

У першому розділі розглянуто актуальність та проблеми використання безпілотних літальних апаратів (БПЛА), що відіграють ключову роль у вирішенні завдань у різних сферах, включаючи військові, комерційні та наукові. Сучасні дослідження демонструють важливість розробки ефективних методів планування, координації та взаємодії в командній роботі, що включає як БПЛА, так і людей. У цьому розділі запропоновано вдосконалений підхід до координації завдань через використання агентних технологій, методів коактивного проектування та планування зі змішаною ініціативою. Особлива увага приділяється адаптації систем до динамічних умов, автоматичній декомпозиції завдань, гнучкому управлінню ресурсами та забезпеченню прозорості у взаємодії. Це дозволяє значно покращити продуктивність командної роботи, особливо в умовах складних і невизначених середовищ.

Висновки по першому розділу. Використання агентних технологій та методів коактивного проектування забезпечує прозорість та узгодженість у взаємодії між людьми та БПЛА, що підвищує ефективність командної роботи.

Запропоновані підходи дозволяють ефективно реагувати на динамічні зміни

умов завдань, зокрема на зміну контексту, появу нових загроз чи змінення можливостей учасників команди.

Інтеграція агентного планувальника значно знижує навантаження на людину-дизайнера, автоматизуючи процес декомпозиції завдань і розподілу ролей.

Запропоновані підходи можуть бути адаптовані для великих гетерогенних команд, що складаються з людей і БПЛА з різними можливостями.

Застосування концепцій спостережливості, передбачуваності та керованості забезпечує кращу поінформованість учасників команди про стан і дії інших.

Результати дослідження демонструють перспективність розвитку інтегрованих підходів до управління командами БПЛА з використанням інноваційних методів, що підвищує ефективність їхньої роботи в складних умовах.

Розділ 2 присвячений методам планування та управління завданнями у командній роботі безпілотних літальних апаратів (БПЛА). Основна увага приділяється оптимізації маршрутів, координації дій та управлінню ресурсами за допомогою різних підходів: централізованого та децентралізованого планування.

Методи централізованого планування. Лінійне програмування використовується для оптимального розподілу завдань та ресурсів. Алгоритми комівояжера допомагають знайти найкоротший маршрут для виконання завдань, з використанням методів перебору, гілок і меж, та наближених алгоритмів. Динамічне програмування дозволяє розділяти задачу на підзадачі, зберігаючи проміжні результати для оптимізації процесу.

Методи децентралізованого планування. Розподілені алгоритми забезпечують автономність дій кожного БПЛА, з можливістю обміну інформацією між агентами. Ройова інтелектуальна поведінка використовує локальну інформацію для колективного вирішення задач на основі таких моделей, як рій частинок, мурашиний алгоритм, зграї птахів та бджолина поведінка. Алгоритми посилення зворотного зв'язку дозволяють БПЛА адаптувати свої дії на основі навчання та досвіду.

Висновки по другому розділу. Централізовані методи забезпечують високу узгодженість дій у команді БПЛА, однак вони мають обмежену масштабованість і залежність від центрального вузла управління.

Децентралізовані підходи підвищують гнучкість і стійкість системи, дозволяючи БПЛА функціонувати навіть у динамічних або неповних умовах.

Ройова інтелектуальна поведінка демонструє ефективність у вирішенні складних задач за рахунок простих правил взаємодії.

Комбінація алгоритмів, таких як динамічне програмування, ройові моделі та алгоритми зворотного зв'язку, дозволяє створювати адаптивні та ефективні системи для командної роботи БПЛА.

Для подальшого вдосконалення командної роботи БПЛА перспективним є використання методів машинного навчання, що дозволить агентам ефективніше адаптуватися до змін середовища та умов виконання завдань.

Розділ 3 висвітлює підхід до коактивного проектування інтерактивних систем, зокрема для організації командної роботи між людьми та безпілотними літальними апаратами (БПЛА). Основні принципи коактивного проектування (спостережливість, передбачуваність та керованість) забезпечують ефективну взаємодію між компонентами системи, дозволяючи людині й БПЛА працювати як єдина команда.

Запропонована структура для командної роботи включає чотири модулі:

- Модуль коактивного проектування аналізує ролі та здібності учасників команди для оптимального розподілу завдань.
- Модуль планування завдань на основі агентів автоматизує процес розподілу завдань та оптимізує взаємодію між агентами.
- Модуль навчання з глибоким підкріпленням реалізує поведінкові моделі, які підвищують автономність БПЛА, зокрема в умовах обмеженого зв'язку.
- Модуль вибору дій зі змішаною ініціативою дозволяє людині-оператору контролювати процес у режимі реального часу через багатофункціональний інтерфейс.

Також розглядається використання ієрархічного планування завдань для забезпечення зрозумілості дій системи для людини. Принципи глибокого підкріпленого навчання дозволяють реалізувати автономну поведінку БПЛА (наприклад, згуртованість у зграї та уникнення небезпек), що сприяє прозорості й керованості системи.

Четвертий розділ присвячений аналізу ієрархічного планувальника завдань, який використовує агентно-орієнтований підхід. Ієрархічний планувальник функціонує в багаторівневій структурі, що складається зі стратегічного, тактичного та оперативного рівнів.

Стратегічний рівень формує загальні цілі та стратегії.

Тактичний рівень розбиває цілі на менші підзадачі.

Оперативний рівень деталізує підзадачі до конкретних інструкцій для агентів.

Передача завдань відбувається зверху вниз (декомпозиція завдань), а зворотний зв'язок здійснюється знизу вгору для коригування дій та адаптації до змін. Планувальник дозволяє ефективно розподіляти ресурси та вирішувати завдання в динамічних умовах.

Також описано формалізацію предметної області, де кожна сутність має набір атрибутів (статичних і динамічних), які використовуються для створення моделей. Декомпозиція завдань представлена як процес, у якому складна задача перетворюється на низку простих дій, що виконуються агентами.

Висновки по четвертому розділу. Ієрархічний планувальник забезпечує чіткість у виконанні стратегічних цілей і гнучкість у реакції на зміни середовища. Використання агентів дозволяє створювати автономні системи, що координують свої дії в рамках спільної стратегії. Ієрархічна структура підтримує масштабування системи без втрати ефективності. Система швидко реагує на нові умови, коригуючи дії як на оперативному, так і на стратегічному рівнях. Чітке визначення сутностей, їх атрибутів та початкових станів створює основу для точного і послідовного моделювання. Завдяки декомпозиції навіть складні завдання розв'язуються поетапно, що полегшує їх розподіл та виконання. Ці характеристики роблять ієрархічний планувальник завдань ефективним інструментом для управління складними системами в динамічних середовищах.

АГЕНТНА КООРДИНАЦІЯ, НЕБЕЗПЕЧНЕ СЕРЕДОВИЩЕ, РОЗПОДІЛ РЕСУРСІВ, ДИНАМІЧНЕ ПЛАНУВАННЯ, БПЛА

Публікації здобувача за темою роботи:

1. Кіреєв А. Ю. (Аксак Н.Г.) Координація агентів у вирішенні завдання пошуку та рятування в небезпечному середовищі Комп'ютерні інтелектуальні системи та мережі. Матеріали XVII Всеукраїнської науково практичної WEB конференції аспірантів, студентів та молодих вчених– Кривий Ріг: Криворізький національний університет, 2024.С. 245-248 <https://sites.google.com/view/kicm/?pli=1>

Використані публікації керівника та співробітників кафедри, що становлять теоретичну базу роботи

2. Н. Г. Аксак, М. В. Кушнар'юв, Мультиагентна система розподіленої торгової фірми. Сучасні інформаційні технології і системи: монографія / В. П. Бурдаєв, Н. Г. Аксак, М. В. Кушнар'юв та ін.; за заг. ред . В. С. Пономаренка. - Харків: Вид. «Стиль-іздат», 2021. - 182 с. <http://repository.hneu.edu.ua/handle/123456789/25920>.

3. Н. Г. Аксак Багатоагентна система електронного навчання. Сучасні інформаційні технології та системи : монографія / Н. Г. Аксак, Л. Е. Гризун, О. В. Щербаков [та ін.] ; за заг. ред. Пономаренка В. С. — Електрон. текстові дан. — Харків: ХНЕУ ім. С. Кузнеця, 2022. — 270 с.

4. Natalia Axak, Mykola Korablyov, Matvii Ushakov. The Development of a Multi-Agent System for Controlling an Autonomous Robot. Proceedings of the 17th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer. Volume I, (ICTERI-2021). – Kherson, Ukraine, 2021. – Vol-3013, pp. 96-105. <https://ceur-ws.org/Vol-3013/20210096.pdf> .

5. Аксак Н., Kushnaryov M., Tatarnykov A.. The development of agent-based learning platform with temporal logic specifications Обробка інформації в системах управління та прийняття рішень. Проблеми та рішення. Монографія. / під наук. ред. проф. В. Вичужаніна – Одеса: НУ «ОМА», 2023 – 358 с. ISBN 978-617-7857-33-3

6. N . Axak, M. Kushnaryov, A.Tatarnykov, The Agent-Based Learning Platform. ICST-2023: XI International Scientific and Practical Conference “Information Control Systems and Technologies”, September 21-23, 2023, Odessa, Ukraine, Vol-3513, pp. 263-275. ONLINE: <http://ceur-ws.org/Vol-3513/>, URN: urn:nbn:de:0074-3513-6

7. Axak N., Kushnaryov M., Tatarnykov A. Agent-driven approach to enhancing

e-learning efficiency. *Advances in Information Control Systems and Technologies: монографія / Н. Аксак, Д. Антонов та ін.; під наук. ред. проф. В.Вичужаніна. – Львів-Торунь :Liha-Pres, 2024 – 380 с.*

8. Ахак N., Kushnaryov M., Tatarnykov A. The Analysis and the Optimization of Learning Trajectories with the Help of Agents. 2024 IEEE 19th International Conference on Computer Sciences and Information Technologies (CSIT), 2024, (16-19 October 2024) (SCOPUS у виданні)

9. Ахак N., Tatarnykov A., Kushnaryov M., Agent-based method of improving the efficiency of the e-learning. ICST-2024: XI International Scientific and Practical Conference “Information Control Systems and Technologies”, September 23-25, 2024, Odessa, Ukraine, Vol-3790, pp. 63-75. ONLINE: <http://ceur-ws.org/Vol-3790/> urn:nbn:de:0074-3790-4

ЗМІСТ

ВСТУП.....	18
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	21
1.1 Аналіз предметної області	21
1.2 Огляд літератури.....	22
1.3 Постановка проблеми.....	26
2 МЕТОДИ ПЛАНУВАННЯ ТА УПРАВЛІННЯ ЗАВДАННЯМИ В КОМАНДНІЙ РОБОТІ З БПЛА	28
2.1 Методи централізованого планування	28
2.1.1 Лінійне програмування	28
2.1.2 Алгоритми комівояжера	29
2.1.3 Динамічне програмування.....	31
2.2 Методи децентралізованого планування	33
2.2.1 Розподілені алгоритми	33
2.2.2 Ройова інтелектуальна поведінка.....	35
2.2.3 Алгоритми посилення зворотного зв'язку.....	37
2.3 Гібридні методи	38
2.3.1 Ієрархічне управління	38
2.3.2 Мультирівнева система.....	39
2.4 Алгоритми оптимізації.....	41
2.4.1 Еволюційні алгоритми	41
2.4.2 Методи рою частинок	42
2.4.3 Методи машинного навчання.....	44
3 МОДЕЛЬ КОМАНДНОЇ РОБОТИ ЛЮДИНИ І БПЛА	46
3.1 Принципи коактивного проектування.....	46
3.2 Структура для командної роботи людей та БПЛА	47
3.4 Аналіз взаємозалежності	52
3.5 Алгоритм взаємодії агентів	53

4 ПЛАНУВАННЯ ЗАВДАНЬ НА БАЗІ АГЕНТІВ	55
4.1 Ієрархічний планувальник	55
4.3 Декомпозиція завдання	60
4.4 Формування плану.....	63
4.4.1 Глибоке навчання стадної поведінки з глибоким підкріпленням	64
4.4.3 Практична реалізація системи командної роботи безпілотних літальних апаратів (БПЛА).....	75
ВИСНОВКИ	82
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	84
ДОДАТОК А Графічний матеріал кваліфікаційної роботи	ПОМИЛКА!
ЗАКЛАДКУ НЕ ВИЗНАЧЕНО.	
ДОДАТОК Б Код програми	ПОМИЛКА! ЗАКЛАДКУ НЕ ВИЗНАЧЕНО.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ШІ – штучний інтелект.

БПЛА – (UAVs - Unmanned Aerial Vehicles) — Безпілотні літальні апарати.

Mixed-initiative – це взаємодія, де обидві сторони можуть активно пропонувати інформацію та ініціювати дії.

DP – Динамічне програмування

ЛП – Лінійне програмування

RL – Reinforcement Learning (Алгоритм посилення зворотного зв'язку)

ML – Машинне навчання

ІАПЗ – ієрархічний агентний планувальник завдань

Агент – обчислювальна сутність, яка має власну автономію і може співпрацювати для вирішення задач.

MULTI-AGENT SYSTEM – система, де кілька обчислювальних сутностей працюють разом, маючи власну автономію, для досягнення спільних цілей.

ВСТУП

Безпілотні літальні апарати (БПЛА) корисні для виконання небезпечних і динамічних завдань, таких як пошуково-рятувальні роботи, спостереження за лісом та антитерористичні операції. Ці завдання можуть бути вирішені краще завдяки співпраці декількох БПЛА під наглядом людини. Однак людині все ще важко відстежувати, розуміти, прогнозувати і контролювати поведінку БПЛА через складність завдань, а також через алгоритми машинного навчання і планування, що використовуються в «чорному ящику». Безпілотні літальні апарати (БПЛА) та мультиагентні системи (МАС) знаходяться на передовій сучасних технологій, які дозволяють виконувати різноманітні завдання в автоматизованому режимі. БПЛА, часто відомі як дрони, використовуються для виконання широкого спектру завдань, від військових операцій до комерційних застосувань, таких як доставка товарів, моніторинг сільськогосподарських угідь, та зйомка. Їхня здатність виконувати місії без прямого втручання людини дозволяє знижувати ризики для операторів та збільшувати ефективність операцій.

Мультиагентні системи, з іншого боку, включають в себе координацію багатьох автономних агентів, які можуть працювати спільно для досягнення певних цілей. Такі системи знаходять застосування у різних сферах, включаючи управління трафіком, розподіл ресурсів, та рятувальні операції. Вони дозволяють розподіляти складні завдання між агентами, що може підвищити ефективність та гнучкість системи в цілому.

Поєднання БПЛА та МАС відкриває нові горизонти для розробки комплексних систем, здатних виконувати завдання, які були б занадто складними або небезпечними для людей. Завдяки розробкам в галузі когнітивних можливостей, планування завдань та навчання з підкріпленням, ці системи стають все більш автономними та ефективними.

Однак, розробка таких систем вимагає глибокого розуміння як технічних аспектів, так і соціальних факторів взаємодії між людьми та машинами.

Використання новітніх методів, таких як коактивне проектування та агентне планування завдань, дозволяє створювати системи, які можуть легко адаптуватися до мінливих умов та забезпечувати надійну та зрозумілу взаємодію з людськими операторами.

Метою роботи є розробка інтелектуальної комп'ютерної системи планування завдань у командній роботі з безпілотними літальними апаратами (БПЛА), яка включає алгоритми оптимізації завдань, координацію дій та інтеграцію даних з різних джерел для забезпечення ефективності виконання місій.

Об'єктом дослідження магістерської кваліфікаційної роботи є інтелектуальні комп'ютерні системи, зокрема система планування завдань в командній роботі з безпілотними літальними апаратами (БПЛА). Це система, що забезпечує ефективну організацію та координацію завдань між учасниками команди (включаючи БПЛА та операторів) для досягнення поставлених цілей.

Предмет дослідження – методи і технології планування завдань у командній роботі, зокрема метод коактивного проектування взаємозалежностей між учасниками команди. А також агентний планувальник завдань, який автоматично декомпозує складні завдання для учасників команди, забезпечуючи ефективну координацію і виконання завдань в умовах взаємозалежності.

Методи дослідження. Для визначення теоретичних аспектів та основних підходів до планування завдань в командній роботі з БПЛА, використовується метод літературного аналізу. Огляд сучасних досліджень та публікацій дозволяє сформулювати загальне уявлення про наявні методи, алгоритми та технології, що використовуються в даній галузі. Використання методів комп'ютерного моделювання для створення симуляцій планування завдань в командній роботі з БПЛА. Це дозволяє отримати експериментальні дані, що демонструють ефективність різних підходів до планування завдань, взаємодії учасників та автоматичної декомпозиції завдань. Для цього використовуються спеціалізовані програмні продукти або створені власні моделі, які забезпечують тестування алгоритмів у контрольованих умовах. Проведення експериментів з реальними або віртуальними системами БПЛА, що дозволяє оцінити роботу запропонованого агентного

планувальника завдань та методу коактивного проектування взаємозалежностей. Це експерименти з виконанням певних завдань у реальних або смодельованих умовах, з оцінкою часу виконання завдань, точності планування та ефективності взаємодії учасників команди. Зібрані під час експериментів дані аналізуються за допомогою статистичних методів, що дозволяє зробити висновки про ефективність запропонованих методів. Для цього використовуються статистичні інструменти для оцінки точності планування, порівняння результатів різних підходів та виявлення закономірностей. Оскільки дослідження пов'язане з агентними системами, використовується метод моделювання агентів для розробки та тестування агентного планувальника завдань. Цей метод дозволяє реалізувати модель взаємодії агентів у команді (включаючи БПЛА), розподілу завдань і автоматичної декомпозиції складних завдань на підзавдання.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Аналіз предметної області

Безпілотні літальні апарати (БПЛА) знайшли широке застосування у військовій, науковій, комерційній та інших сферах діяльності, таких як моніторинг навколишнього середовища, картографування, доставлення вантажів, патрулювання та інші. Зі збільшенням їхнього використання виникає потреба в удосконаленні методів планування та координації завдань, які покладаються на БПЛА в межах командної роботи. В умовах колективного виконання завдань кількома БПЛА або людьми, які взаємодіють з ними, важливим завданням є ефективне планування та розподіл завдань, що враховують не тільки технічні, а й людські ресурси. Кожен учасник команди (як оператор, так і сам БПЛА) має виконувати свою частину роботи відповідно до загального плану, що потребує складних алгоритмів координації та делегування завдань.

Використання агентних технологій дозволяє створювати динамічні системи, де учасники можуть самостійно приймати рішення, аналізувати взаємозалежності та автоматично розподіляти завдання. Такий підхід дає змогу підвищити ефективність планування, зменшити людське втручання в процес прийняття рішень та забезпечити адаптивність системи до змінюваних умов. Однак, існуючі підходи не завжди здатні забезпечити оптимальний рівень координації та автоматизації, що робить цю тему актуальною для подальших досліджень.

У командній роботі важливо враховувати взаємозалежність між учасниками, що здійснюють планування та виконання завдань. Метод коактивного проектування дозволяє правильно узгодити дії між членами команди, що особливо важливо в умовах роботи з БПЛА, де точність виконання завдань залежить від синхронізації кількох учасників (як людей, так і апаратів). Розробка таких методів є важливим кроком для покращення ефективності планування в складних системах.

Складні завдання, які можуть бути поставлені перед командою БПЛА, потребують їх автоматичної декомпозиції на дрібніші підзавдання. Це дозволяє значно знизити рівень складності управління завданнями і підвищити швидкість їх виконання. Проте, існуючі методи декомпозиції часто не враховують динамічних змін у виконанні завдань або взаємозалежностей між підзавданнями, що є важливою проблемою для розвитку таких систем.

Таким чином, робота є актуальною, оскільки вона дозволяє вирішити низку проблем у сфері планування та координації завдань для команд БПЛА, підвищує ефективність управління такими системами через застосування новітніх агентних технологій та методів автоматизації, а також сприяє розвитку більш гнучких і адаптивних підходів до планування завдань у командній роботі.

1.2 Огляд літератури

Безпілотні літальні апарати (БПЛА) стали важливим інструментом у місіях з ліквідації наслідків стихійних лих і військових операціях. Такі ситуації зазвичай небезпечні, брудні, динамічні та складні для людей. Натомість рій БПЛА може бути більш надійним для виконання великомасштабних завдань через випадкові збої, такі як пошкодження обладнання або відключення зв'язку. Багато роботи було зроблено в галузі рійової робототехніки [1, 2] і мультиагентних систем (МАС)[3] з точки зору сприйняття [4], управління [5], планування [6], прийняття рішень [7], навчання [8], комунікації [9] і тощо. Однак, гнучка і масштабована команда БПЛА, що співпрацюють між собою, може вирішувати складні завдання в невизначеному і динамічному середовищі.

Людина все ще відіграє важливу роль у нагляді за БПЛА, оскільки вони навряд чи можуть досягти повної автономії для самостійного вирішення завдань без втручання людини [10]. Крім того, повна автономія не є бажаною для автономних роботів з огляду на етичні питання [11]. У таких випадках людина все одно необхідна. Наприклад, доведено, що вона ефективна для підвищення продуктивності алгоритмів комп'ютерного зору для розпізнавання об'єктів і відстеження цілей у складних умовах

деформації, оклюзії або розмиття при русі [12, 13, 14]. У літературі запропоновано безліч інтерактивних систем «людина-робот», включаючи телеуправління [15], диспетчерське управління [16], змішану ініціативну взаємодію [17], адаптивну автономію [18], регульовану автономію [19], спільне управління [20] і коактивне проектування [21] та ін. Однак багато з них орієнтовані на автономію, тобто на підвищення рівня автономії (LOA) членів команди до якомога вищого рівня. Це спричиняє проблему непрозорості, яка полягає в тому, що члени команди навряд чи можуть підтримувати достатню обізнаність про стан і дії інших для ефективної командної роботи [22]. У цій роботі розроблено автономного лояльного напарника зі зрозумілим плануванням і можливостями навчання для прозорості командної роботи людини і БПЛА.

На противагу цьому, коактивне проектування [21] забезпечує прозорість командної роботи людини і робота, аналізуючи відносини взаємозалежності між членами команди на основі їхніх здібностей, необхідних для вирішення поставлених завдань. Модель коактивної системи орієнтована на командну роботу і використовує концепцію спостережливості, передбачуваності та керованості (OPD) як орієнтир для проектувальників, щоб визначити вимоги до злагодженої командної роботи. Теорія коактивного проектування виявилася корисною в ряді завдань, включаючи спільну навігацію людини і БПЛА в захищених приміщеннях [21], пошук і доставку в симуляційних середовищах Blocks World for Teams (BW4T), маніпулювання роботами [22] в DARPA Robotics Challenge [23], а також спільну оцінку пошкоджень людиною і БПЛА в реальних умовах [21]. На відміну від оригінальної роботи з коактивного дизайну, яка просто використовує кілька кольорів для представлення рівнів можливостей для вирішення завдання або забезпечення підтримки завдання, реальні значення можуть бути використані для представлення можливостей і розрахунку ймовірності успіху завдання з урахуванням ситуативних умов, таких як ворожі загрози в задачі спостереження за допомогою БПЛА людини [22].

Однак у методу коактивного проектування є кілька недоліків. По-перше, він вимагає від досвідченої людини-дизайнера декомпозиції завдання на виконувани підзадачі, а потім розподілу підзадач між здібними членами команди заздалегідь

спланованим способом. Цей процес проектування може вимагати декількох раундів ітерацій і накладає велике навантаження на людину-проектувальника. По-друге, він навряд чи може впоратися з динамічними ситуаціями, коли можливості члена команди змінюються або змінюється контекст завдання. По-третє, він зазвичай має справу з невеликою людино-робочою командою, що складається з одного робота та однієї людини [21, 22, 23]. Іншими словами, він не дуже добре масштабується для координації більшої команди роботів і людей. В останньому випадку відносини взаємозалежності між членами команди стануть дуже складними. У цій роботі автори вирішують першу проблему шляхом впровадження агентного планувальника завдань для автоматичної генерації планів спільної роботи людини і БПЛА. Також, розглядається поява неочікуваних подій у динамічному середовищі завдань, а також більшу і складнішу команду людей і БПЛА, що складається з гетерогенних агентів з різними можливостями.

Також використовуючи децентралізований підхід, завдання реагування на катастрофу моделюється за допомогою частково спостережуваного Марківського процесу прийняття рішень (POMDP), і оптимальне спільне планування шляхів людей і БПЛА може бути розраховане шляхом вирішення POMDP на кожному часовому кроці. Однак ці підходи не орієнтовані на командну роботу і не враховують вимоги OPD (спостережливість, передбачуваність і керованість) [21] для прозорості командної роботи.

На відміну від методів планування, які генерують плани, пов'язані із законами управління цілями завдання або траєкторіями, навчання з підкріпленням (RL) забезпечує послідовну структуру прийняття рішень, в якій оптимальні політики вибору дій можуть бути вивчені через взаємодію агентів з навколишнім середовищем методом спроб і помилок. Зокрема, було показано, що глибоке навчання з підкріпленням (Deep Reinforcement Learning, DRL) і мультиагентне навчання з підкріпленням (Multi-Agent Reinforcement Learning, MARL) дозволяють досягти продуктивності на рівні людини в декількох областях завдань.

Сучасна тематика охоплює дослідження, спрямовані на розробку та впровадження мультиагентних систем для управління складними розподіленими

процесами, такими як: управління розподіленими комерційними мережами [25], оптимізація навчальних траєкторій в електронному навчанні [26, 29 – 33], контроль автономних роботів із використанням мультиагентного підходу [27], використання темпоральної логіки для проектування навчальних платформ [28].

Висновки по результатах огляду літератури:

Безпілотні літальні апарати (БПЛА) набули значної популярності в рятувальних місіях та військових операціях через свою здатність працювати в небезпечних, динамічних і складних умовах. Завдяки їх здатності діяти у вигляді рою, БПЛА можуть забезпечувати надійність і ефективність у вирішенні завдань навіть у випадку збоїв або відключень зв'язку.

Незважаючи на розвиток автономії, людина продовжує відігравати важливу роль у нагляді за БПЛА. Повна автономія ще не досягнута через етичні питання та обмеження технологій, що вимагають залучення людини для забезпечення ефективного виконання завдань, зокрема, для покращення продуктивності алгоритмів комп'ютерного зору.

Теорія коактивного проектування сприяє розвитку прозорих взаємодій між людьми і роботами, забезпечуючи більш ефективну командну роботу через аналіз взаємозалежностей між агентами. Однак існують обмеження, пов'язані з необхідністю великого навантаження на людину-дизайнера та труднощами в обробці динамічних змін у задачах.

Метод коактивного проектування не завжди ефективний для великих і складних команд з гетерогенними агентами, оскільки не враховує динамічних змін у можливостях агентів та зміну контексту завдань. Це вимагає використання автоматичних систем планування для полегшення процесу проектування.

У роботі запропоновано агентний планувальник завдань, який автоматично генерує плани для співпраці людини і БПЛА, враховуючи складні завдання, ресурси та часові обмеження. Це дозволяє уникнути недоліків традиційних методів коактивного проектування, підвищуючи ефективність роботи в реальних умовах.

Використання змішаної ініціативи для координації між людьми та агентами дозволяє досягти гнучкої автономії. Навчання з підкріпленням (RL) та мультиагентне

навчання з підкріпленням (MARL) також сприяють покращенню спільної діяльності агентів, однак питання прозорості і зрозумілості їх рішень залишаються актуальними.

Запропонована нова структура співпраці на основі вимог спостережливості, передбачуваності та керованості (OPD) забезпечує кращу обізнаність членів команди та покращує взаємодію між людьми і БПЛА порівняно з попередніми методами.

Розроблений агентний планувальник завдань здатний автоматично розбивати складні завдання на підзадачі з урахуванням ресурсних обмежень і соціальних правил, що підвищує ефективність спільної роботи людини і БПЛА.

Таким чином, розробка та впровадження агентних систем планування і координації завдань сприяє покращенню командної роботи між людьми та БПЛА, забезпечуючи гнучкість, прозорість та ефективність у складних, динамічних середовищах.

1.3 Постановка проблеми

Наведені аспекти вказують на важливі проблеми, які потребують вирішення:

Невизначеність у виборі та оптимальному розподілі завдань серед учасників команди БПЛА.

Визначення ефективних методів для інтеграції даних з БПЛА для прийняття рішень в режимі реального часу.

Метою цієї роботи є розробка інтелектуальної комп'ютерної системи планування завдань у командній роботі з безпілотними літальними апаратами (БПЛА), яка включає алгоритми оптимізації завдань, координацію дій та інтеграцію даних з різних джерел для забезпечення ефективності виконання місій.

Для досягнення цієї мети необхідно вирішити наступні завдання:

- Оцінити існуючі методи планування та управління завданнями в командній роботі з БПЛА.
- Розробити алгоритми для інтеграції даних з БПЛА та забезпечення злагодженого виконання завдань.

– Створити структуру інтелектуальної системи для планування та моніторингу виконання завдань.

2 МЕТОДИ ПЛАНУВАННЯ ТА УПРАВЛІННЯ ЗАВДАННЯМИ В КОМАНДНІЙ РОБОТІ З БПЛА

Командна робота безпілотних літальних апаратів (БПЛА) базується на координації, розподілі завдань і ефективному управлінні ресурсами. Для цього використовуються різні методи, які забезпечують оптимізацію маршрутів, виконання місій і взаємодію між апаратами.

2.1 Методи централізованого планування

Ці методи передбачають наявність центрального вузла управління, який відповідає за планування завдань і координацію дій усіх БПЛА

2.1.1 Лінійне програмування

Цей метод використовується для оптимального розподілу завдань та ресурсів між апаратами.

Лінійне програмування (ЛП) – це математичний метод оптимізації, який використовується для знаходження найкращого рішення у задачах розподілу обмежених ресурсів. Він ефективно застосовується для оптимального планування завдань і ресурсів між безпілотними літальними апаратами (БПЛА), особливо в умовах чітко визначених обмежень та критеріїв.

Припустимо, є три БПЛА, які мають виконати завдання у трьох різних точках. Заряд батареї обмежений, а завдання мають пріоритети. Використання ЛП допомагає розподілити завдання так, щоб мінімізувати час польоту і максимально використати доступні ресурси, враховуючи пріоритетність завдань.

Цільова функція визначає мету оптимізації, наприклад, мінімізацію часу виконання завдань, максимізацію покриття території чи мінімізацію витрат енергії.

$$\text{Maximize (або Minimize): } Z = c_1 x_1 + c_2 x_2 + \dots + c_n x_n, \quad (2.1)$$

де Z – цільова функція, c_i – коефіцієнти, що відображають вагомість кожного завдання, x_i – змінні рішення.

Обмеження задаються у вигляді системи лінійних рівнянь чи нерівностей, які враховують наявні ресурси (час, енергія, зона покриття):

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \leq b_1, \quad (2.2)$$

де a_i – коефіцієнти впливу змінної x_i на обмеження, b_j – граничне значення ресурсу.

Усі змінні мають бути невід'ємними, оскільки вони відображають реальні фізичні параметри (час, кількість завдань, витрати): $x_i \geq 0$.

Визначення цільової функції:

- Максимізація кількості виконаних завдань.
- Мінімізація часу роботи всього флоту БПЛА.

Обмеження можуть включати:

- Доступні ресурси БПЛА (заряд батареї, час у повітрі, вантажопідйомність).
- Часове вікно виконання завдань.
- Просторові обмеження (зони, в яких працюють БПЛА).

Задачу лінійного програмування можна розв'язати за допомогою таких методів: симплекс-метод – ефективний для середніх і великих задач та метод внутрішніх точок підходить для задач з великою кількістю змінних.

Автоматизованими інструментами є MATLAB, Python (бібліотека PuLP або SciPy), IBM CPLEX.

Рішення показує оптимальний розподіл завдань між БПЛА з урахуванням ресурсів і обмежень.

Перевагами методу є точність, о забезпечує оптимальне рішення для чітко заданих умов; гнучкість тобто легко адаптується для різних цілей і обмежень та ефективність, яка дозволяє розв'язувати великі задачі розподілу в реальному часі.

2.1.2 Алгоритми комівояжера

Алгоритм комівояжера є класичним підходом для вирішення задачі маршрутизації, коли необхідно знайти найкоротший шлях, який дозволяє відвідати задану кількість точок та повернутися до початкової. Цей алгоритм особливо ефективний для оптимізації маршрутів виконання завдань БПЛА, які повинні мінімізувати витрати часу, енергії або відстані. Алгоритми комівояжера підходять для визначення оптимальних маршрутів виконання завдань, особливо у випадках, коли БПЛА мають відвідати кілька точок.

Суть задачі комівояжера. Припустимо, що ми маємо:

- набір точок (завдань), які БПЛА має відвідати;
- метричний граф, де вершини – це точки, а ребра представляють відстані, час або витрати між точками.

Необхідно знайти оптимальний маршрут, який дозволить відвідати всі точки один раз і мінімізувати загальні витрати (наприклад, відстань або енергоспоживання).

Формалізація задачі виглядає так.

Побудова графа, де кожна точка (завдання) – це вершина, а відстань або витрати між точками – вага ребра.

Цільова функція: мінімізувати загальну довжину маршруту:

$$\text{Minimize } Z = \sum_{i=1}^n d(x_i, x_{i+1}), \quad (2.3)$$

де $d(x_i, x_{i+1})$ – відстань між точками x_i та x_{i+1} .

Методами розв'язання є задача комівояжера з NP-складністю, тобто точний розв'язок для великої кількості точок вимагає значного часу. Тому застосовують різні підходи:

Перебір – знаходження всіх можливих маршрутів і вибір найкоротшого. Підходить для невеликої кількості точок.

Метод гілок і меж – оптимізація перебору, що відкидає явно неефективні маршрути.

Метод найближчого сусіда починається з однієї точки та додає до маршруту

найближчу ще не відвідану точку.

Метод двохстороннього пошуку – оптимізація на основі перевірки та покращення маршруту шляхом виключення перехресть.

Генетичні алгоритми використовують принципи природного відбору для пошуку оптимального маршруту.

Алгоритм мурашиної колонії імітує поведінку мурах, які знаходять оптимальні шляхи між джерелами їжі та гніздом.

Після обчислення маршрут перевіряється на відповідність вимогам, наприклад: врахування зон обмеження польотів, доступного заряду батареї або часу виконання завдань.

До особливостей застосування в роботі БПЛА відноситься таке. Алгоритм ефективно вирішує задачі для невеликої кількості точок; для великих флотів БПЛА можуть знадобитися наближені рішення. Легко адаптується до специфіки завдань, таких як додаткові обмеження на час перебування в точці або пріоритетність завдань. Алгоритм можна комбінувати з іншими підходами, наприклад, лінійним програмуванням, для врахування додаткових ресурсів.

У якості приклада можна припустити, що БПЛА має доставити посилки в 5 різних точок. Використовуючи алгоритм комівояжера формують граф відстаней між точками (наприклад, на основі GPS-координат). За допомогою алгоритму (наприклад, методу мурашиної колонії) обчислюють найкоротший маршрут. Результатом є маршрут, який мінімізує загальні витрати енергії та часу, враховуючи технічні обмеження БПЛА.

2.1.3 Динамічне програмування

Динамічне програмування (DP) – це метод розв'язання задач, що передбачає поділ на підзадачі, розв'язки яких зберігаються і використовуються для знаходження оптимального рішення всієї задачі. DP ідеально підходить для задач із змінними умовами, такими як зміна пріоритетів завдань у командній роботі з БПЛА. Застосовується для задач, які передбачають змінні умови, наприклад, зміну

пріоритетів завдань.

Динамічне програмування використовується, коли:

1. Задачу можна розділити на підзадачі, що розв'язуються незалежно.
2. Існує можливість повторного використання розв'язків підзадач.
3. Умови змінюються, наприклад, пріоритети завдань, обмеження на ресурси, час виконання.

Замість прямого перебору всіх можливих варіантів (що є обчислювально дорого), DP запам'ятовує вже обчислені розв'язки та оптимізує наступні етапи.

Етапи динамічного програмування

Етап 1. Формулювання станів.

Визначається, як описати кожен стан системи, наприклад:

- Які завдання вже виконані.
- Скільки ресурсів залишилося.
- Поточний пріоритет завдань.

Етап 2. Рекурсивна залежність.

Встановлюється зв'язок між поточним станом і попередніми станами, щоб використовувати раніше обчислені розв'язки:

$$F(state) = \min/\max\{cost(state', task) + F(state')\}, \quad (2.4)$$

де $state'$ – попередній стан, а $cost(state', task)$ – вартість виконання завдання в поточному стані.

Етап 3. Збереження результатів.

Для уникнення повторних обчислень усі проміжні результати зберігаються в таблиці (memoization або tabulation).

Етап 4. Відновлення оптимального розв'язку.

Після заповнення таблиці визначається оптимальний шлях виконання завдань, наприклад, які завдання виконати першими.

У контексті командної роботи з БПЛА метод DP дозволяє враховувати змінні умови, такі як:

Кожне завдання може отримати нову вагу, і DP перебудовує оптимальний розв'язок.

Якщо під час виконання завдань змінюється заряд батареї або з'являються нові обмеження, DP дозволяє адаптувати план.

Зміна точок призначення або додавання нових завдань може бути інтегровано без перерахунку всіх можливих варіантів.

Наприклад. Флот БПЛА повинен виконати кілька завдань з різними пріоритетами, маючи обмежений заряд батареї.

Завдання 1. Формулювання станів.

- $state = (task_done, battery_left)$, де $task_done$ – виконані завдання,
- а $battery_left$ – залишок заряду.

Завдання 2. Рекурсивна залежність.

Якщо $task_i$ виконується, новий стан обчислюється як:

$$F(task_done \cup \{task_i\}, battery_left - cost(task_i)) = \max \{ priority(task_i) + F(next_state) \}.$$

Завдання 3. Заповнення таблиці.

Зберігаються розв'язки для кожного стану $F(state)F(state)F(state)$, щоб уникнути повторних обчислень.

Завдання 4. Визначення оптимального розв'язку.

Знаходиться послідовність завдань, яка забезпечує максимальний сумарний пріоритет із врахуванням доступного заряду.

2.2 Методи децентралізованого планування

Ці методи передбачають автономне прийняття рішень кожним БПЛА на основі локальної інформації.

2.2.1 Розподілені алгоритми

В цьому випадку метод децентралізованого планування базується на використанні розподілених алгоритмів, де кожен безпілотний літальний апарат

(БПЛА) самостійно планує власні дії, обмінюючись інформацією з іншими. Цей підхід дозволяє створювати гнучкі, масштабовані та стійкі системи командної роботи БПЛА, які можуть ефективно функціонувати навіть за умов неповної або динамічної інформації кожен апарат планує власні дії, обмінюючись інформацією з іншими.

У децентралізованому плануванні відсутній єдиний центр управління, натомість кожен агент (БПЛА) виконує планування локально, з урахуванням:

- Інформації про власний стан: заряд батареї, доступні ресурси, поточне завдання.

- Обміну інформацією з іншими БПЛА: місце розташування, виконані завдання, зміни умов.

Цей підхід зменшує залежність від центрального вузла та підвищує стійкість системи до збоїв окремих компонентів.

Кожен БПЛА самостійно розробляє план дій на основі локально доступної інформації. Агент обмінюється інформацією з іншими БПЛА через мережу зв'язку (наприклад, бездротові протоколи). Це дозволяє координувати дії та уникати конфліктів. Завдання розподіляються динамічно, кожен БПЛА бере на себе те, що найкраще відповідає його можливостям і стану. Система здатна адаптуватися до змінних умов, таких як поява нових завдань, вихід з ладу одного з БПЛА або зміна ресурсів.

Етапи роботи розподіленого алгоритму.

Етап 1. Ініціалізація. Кожен БПЛА отримує початкову інформацію про свої завдання, ресурси та умови роботи.

Етап 2. Локальне планування. На основі власних ресурсів і найближчих завдань кожен апарат створює локальний план дій.

Етап 3. Обмін інформацією. Агент передає свої дані іншим БПЛА (наприклад, інформацію про виконання завдань або маршрут). Це дозволяє узгодити дії між усіма учасниками.

Етап 4. Координація. БПЛА погоджують маршрути, щоб уникнути конфліктів (наприклад, двоє агентів не повинні виконувати одне й те саме завдання).

Етап 5. Адаптація. Якщо умови змінюються (наприклад, один з агентів не може

завершити завдання), інші БПЛА перерозподіляють свої плани відповідно до нових даних.

Приклад застосування децентралізованого планування.

Флот із 5 БПЛА повинен обстежити ділянку, кожен апарат отримує координати кількох точок для інспекції. Кожен БПЛА визначає точки, які знаходяться в його зоні досяжності. БПЛА обмінюються інформацією про точки, які вони планують обстежити, щоб уникнути дублювання. Один із БПЛА змінює свій план через повідомлення від іншого, який ближче до деяких точок. Якщо один з апаратів виходить з ладу, інші беруть на себе його завдання.

Розподілені алгоритми в децентралізованому плануванні забезпечують ефективну роботу команд БПЛА в умовах, де необхідна гнучкість, стійкість до збоїв і здатність адаптуватися до змін середовища або завдань. Вони особливо корисні для задач із динамічними умовами, такими як розвідка, доставка чи пошук і порятунок.

2.2.2 Ройова інтелектуальна поведінка

Ройова інтелектуальна поведінка – це метод управління колективною діяльністю безпілотних літальних апаратів (БПЛА), заснований на імітації поведінки природних систем, таких як мурах, бджіл чи птахів. Основою такого підходу є взаємодія між агентами на базі простих правил, що дозволяє системі досягати високого рівня узгодженості та ефективності без центрального управління. Дії кожного апарата базуються на простих правилах взаємодії, що імітують поведінку природних систем, наприклад, мурах чи бджіл.

Кожен агент (БПЛА) діє автономно, приймаючи рішення на основі локальної інформації, але координує свої дії з іншими завдяки обміну інформацією. Сукупність таких взаємодій створює "роєву" поведінку, яка дозволяє групі агентів вирішувати складні задачі, наприклад:

- Планування оптимальних маршрутів.
- Виконання завдань із дослідження територій.
- Уникнення перешкод і колективний пошук ресурсів.

Принципи ройової поведінки полягає у відсутності єдиного центру управління, кожен агент діє автономно. Агенти приймають рішення на основі інформації від своїх "сусідів" та з довкілля. Дії агентів базуються на базових принципах, таких як уникнення зіткнень, слідування за сусідом, чи пошук цілей. Система може реагувати на зміни в реальному часі, наприклад, зміну середовища чи вихід із ладу одного з агентів.

До основних моделей ройової поведінки відносяться:

Модель рою частинок, яка використовується для пошуку оптимальних рішень. Кожен агент оновлює свою позицію у просторі, враховуючи як власний досвід, так і досвід сусідів.

Мурашиний алгоритм, у якому агент (БПЛА) залишає "сліди" у середовищі (аналог феромонів у мурах), які допомагають іншим агентам знаходити оптимальні маршрути.

Імітація поведінки зграї птахів, де агенти слідують за трьома простими правилами: уникнення зіткнень, вирівнювання з сусідами, і прагнення до центру групи.

Бджолина модель в якій агенти (аналог бджіл) досліджують середовище, повертаючись до "вулика", щоб обмінятися інформацією про знайдені ресурси.

Робота ройового підходу складається з таких етапів:

Етап 1. Ініціалізація. Кожен агент отримує початкові параметри, такі як координати, енергію або цільові завдання.

Етап 2. Взаємодія. БПЛА обмінюються інформацією зі "сусідами", використовуючи локальні протоколи зв'язку.

Етап 3. Адаптація. Кожен агент змінює свій маршрут або дії залежно від інформації, отриманої від інших, чи змін середовища.

Етап 4. Колективне досягнення мети. Через координовані дії всі агенти виконують завдання ефективніше, ніж у випадку ізольованих дій.

Ройова інтелектуальна поведінка є ефективним підходом для координації дій великих груп БПЛА, особливо у складних, динамічних середовищах. Використання цього методу дозволяє досягти високого рівня гнучкості, адаптивності та

автономності системи.

2.2.3 Алгоритми посилення зворотного зв'язку

Алгоритми посилення зворотного зв'язку забезпечують адаптивне навчання БПЛА на основі попереднього досвіду.

Алгоритм посилення зворотного зв'язку (Reinforcement Learning, RL) – це метод машинного навчання, який дозволяє безпілотним літальним апаратам (БПЛА) навчатися шляхом взаємодії з довкіллям. Навчання відбувається за рахунок отримання зворотного зв'язку у вигляді нагороди (reward) або штрафу (penalty) за виконання певних дій. Головною метою є оптимізація стратегії (policy), яка максимізує кумулятивну винагороду в процесі виконання завдань.

Основними компонентами RL є Агент БПЛА, який виконує дії та взаємодіє з середовищем; середовище – об'єкт, з яким агент взаємодіє (наприклад, територія для патрулювання чи виконання місій); стан (S) – опис поточного стану середовища, який сприймає агент; дія (A) – вибір агента щодо того, що робити в поточному стані; нагорода (R) – зворотний зв'язок про ефективність дії агента.

Принцип роботи алгоритму RL полягає у такому: агент починає роботу з випадковою стратегією дій; агент виконує дію, змінює стан середовища та отримує нагороду; агент аналізує отриману винагороду та коригує свою стратегію; з кожною новою ітерацією стратегія стає кращою, адже агент накопичує досвід.

Для реалізації методів RL для БПЛА можна використовувати такі алгоритми:

Q-Learning. Алгоритм побудови таблиці, де зберігається значення нагороди для кожного стану і дії. Агент оновлює ці значення на основі отриманого зворотного зв'язку. Використовується для задач із дискретними станами.

Deep Reinforcement Learning (DRL). Комбінація RL із глибинним навчанням, де нейронна мережа замінює таблицю Q-значень. Підходить для складних та динамічних середовищ із великим числом станів. Агент навчається на основі попередніх даних і прогнозує найкращі дії в нових ситуаціях.

Actor-Critic метод. Розподіляє навчання між двома компонентами:

Actor (актор): генерує дії на основі поточного стану.

Critic (критик): оцінює якість цих дій.

Цей алгоритм забезпечує швидше та точніше навчання.

RL дозволяє БПЛА оптимально обирати маршрути, враховуючи змінні умови (наприклад, зміну погоди чи виявлення перешкод). Алгоритм адаптивно навчається ефективно використовувати обмежені ресурси, такі як енергія чи час польоту. У групі БПЛА кожен агент може навчатися координувати свої дії з іншими, щоб виконувати завдання максимально ефективно. Наприклад, у випадку виявлення загрози чи поломки один із БПЛА адаптує свої дії, щоб компенсувати втрату іншого.

2.3 Гібридні методи

2.3.1 Ієрархічне управління

Ієрархічне управління є гібридним методом, який поєднує переваги централізованого і децентралізованого підходів до управління групою безпілотних літальних апаратів (БПЛА). У цій моделі управління використовується багаторівнева структура, де центральний вузол відповідає за визначення загальної стратегії, а окремі БПЛА виконують тактичні завдання на основі локальних умов і отриманих інструкцій.

Центральний вузол відповідає за загальну стратегію, а окремі апарати – за тактичні дії.

Центральний вузол (контролер) відповідає за стратегічне планування, глобальний розподіл ресурсів і завдань. Аналізує ситуацію на основі даних від усіх БПЛА та коригує загальну стратегію в реальному часі.

Локальні агенти (БПЛА) виконують тактичні дії, орієнтуючись на локальні умови та поточні задачі. Можуть автономно приймати рішення в межах своєї зони відповідальності. Регулярно обмінюються інформацією з центральним вузлом і між собою для координації дій.

Етапи реалізації ієрархічного управління.

Етап 1. Стратегічне планування. Центральний вузол отримує дані про загальні цілі місії (наприклад, пошук об'єктів або моніторинг території) і визначає пріоритети, ресурси та маршрути.

Етап 2. Розподіл завдань. Завдання розподіляються між БПЛА на основі їхніх можливостей, позицій і поточного стану (заряд батареї, навантаження тощо).

Етап 3. Виконання завдань. Окремі БПЛА автономно виконують свої завдання, адаптуючись до змінних умов (наприклад, обхід перешкод або реагування на нові події).

Етап 4. Зворотний зв'язок. Локальні агенти надсилають дані до центрального вузла про виконання завдань і стан середовища. Центральний вузол використовує ці дані для оновлення стратегії та перерозподілу завдань у разі потреби.

Ієрархічне управління забезпечує баланс між централізованим стратегічним плануванням і локальною автономією БПЛА. Це робить його ефективним для виконання складних задач у динамічних середовищах, що потребують як глобальної координації, так і оперативної адаптації на місцях.

2.3.2 Мультирівнева система

Мультирівнева система – це підхід до планування, який розділяє процес управління завданнями на кілька рівнів із різною деталізацією, масштабом і функціональністю. Такий підхід дозволяє оптимізувати розподіл завдань і ресурсів між групою БПЛА, забезпечуючи ефективність і адаптивність у складних динамічних умовах. Планування виконується на кількох рівнях, наприклад, стратегічний рівень (централізований) і тактичний рівень (децентралізований).

Система розбивається на кілька рівнів, кожен із яких відповідає за певний аспект планування:

На стратегічному рівні загальне планування місії на основі глобальних цілей і умов.

На операційному рівні плануються дії на середньому масштабі з урахуванням ресурсів і зон відповідальності.

На тактичному рівні дійснюється деталізоване планування окремих завдань і локальних дій для кожного БПЛА.

Кожен рівень фокусується на своїх задачах, забезпечуючи узгодженість між загальною стратегією і локальними діями. Дані про виконання завдань і зміни в середовищі передаються між рівнями для коригування планів.

Етапи мультирівневого планування

Стратегічне планування. Визначення глобальних цілей місії (патрулювання, моніторинг, рятувальні операції тощо). Розподіл зон відповідальності між групами БПЛА. Оцінка ресурсів і часу, необхідного для виконання місії.

Операційне планування. Формування підзадач для кожної групи або окремого БПЛА. Розподіл завдань з урахуванням пріоритетів, обмежень і поточних умов. Координація між групами БПЛА для уникнення конфліктів і дублювання дій.

Тактичне планування. Побудова маршрутів і визначення детальних дій кожного БПЛА. Оперативне реагування на зміни в середовищі (наприклад, поява нових перешкод чи цілей).

Мультирівневе планування легко адаптується до збільшення кількості БПЛА або ускладнення місій. Кожен рівень може використовувати свої алгоритми планування, що полегшує розробку та налаштування системи. Нижчі рівні можуть самостійно адаптуватися до локальних змін, зберігаючи узгодженість із загальною стратегією.

Наприклад, на стратегічному рівні визначаються зони спостереження (ліси, водойми, сільськогосподарські угіддя).

Операційний рівень розподіляє групи БПЛА між зонами.

На тактичному рівні кожен БПЛА виконує маршрут, збираючи дані про температуру, вологість, забруднення тощо.

Пошуково-рятувальні операції пов'язані з визначенням району пошуку і встановлює загальні цілі. Розподіляє зони пошуку між командами БПЛА. Тактичний рівень: Кожен БПЛА сканує виділену область і передає дані про виявлені об'єкти.

Мультирівнева система забезпечує баланс між стратегічним управлінням і локальною автономією, що робить її ефективним підходом для складних задач у

динамічних середовищах. Вона широко використовується в задачах, пов'язаних із координацією великих груп БПЛА, забезпечуючи їхню ефективність і узгодженість дій.

2.4 Алгоритми оптимізації

2.4.1 Еволюційні алгоритми

Еволюційний алгоритм — це метод оптимізації, заснований на принципах природної еволюції, таких як відбір, мутація, кросовер та адаптація. Він ефективно використовується для вирішення складних задач розподілу ресурсів і завдань, де кількість можливих рішень є дуже великою, а точний перебір усіх варіантів неможливий. Використовуються для складних завдань розподілу, які мають велику кількість можливих рішень.

До основних етапів еволюційного алгоритму відносяться

Ініціалізація популяції. Генерується початкова популяція можливих рішень (хромосом). Кожна хромосома представляє потенційне рішення задачі, наприклад, розподіл завдань між БПЛА.

Оцінка (функція придатності). Для кожного індивіда в популяції обчислюється значення функції придатності, яка оцінює якість рішення. Функція враховує різні критерії, наприклад, мінімізацію часу виконання завдань, зменшення витрат енергії або балансування навантаження між апаратами.

Селекція. Відбираються найкращі індивіди з поточної популяції для створення наступного покоління. Алгоритми відбору можуть бути різними, наприклад, рулетка (пропорційно значенню придатності) або турнірний відбір.

Кросовер (рекомбінація). Відбірні індивіди об'єднуються (схрещуються) для створення нових рішень. Цей процес дозволяє обмінюватися інформацією між успішними рішеннями, формуючи нові, потенційно кращі хромосоми.

Мутація. Внесення випадкових змін у деякі хромосоми для збільшення різноманітності популяції. Це допомагає уникнути «застрявання» в локальних

оптимумах.

Заміна поколінь. Новостворені індивіди формують нове покоління, яке замінює попереднє.

Процес повторюється до досягнення критеріїв зупинки (наприклад, досягнення заданої кількості поколінь або прийнятного рівня функції придатності).

Еволюційні алгоритми широко використовуються для складних задач розподілу завдань і маршрутів у командній роботі БПЛА.

Приклад застосування для БПЛА. Визначення найкоротших і найефективніших шляхів виконання завдань для кожного дрона з урахуванням обмежень (батарея, час). Розподіл завдань між апаратами з урахуванням їхніх можливостей, наприклад, вантажопідйомності або швидкості. Реагування на непередбачувані зміни, наприклад, появу нових завдань або перешкод. Функція придатності для задачі розподілу завдань:

$$F = w_1 \cdot T_{min} + w_2 \cdot E_{min} + w_3 \cdot B_{bal} , \quad (2.5)$$

де T_{min} – мінімізація часу виконання завдань;

E_{min} – зменшення енергоспоживання;

B_{bal} – балансування навантаження між БПЛА;

w_1, w_2, w_3 – вагові коефіцієнти, що задають пріоритети.

Еволюційний алгоритм є потужним інструментом для вирішення складних задач у динамічних середовищах. Його використання в командній роботі БПЛА дозволяє ефективно розподіляти завдання, оптимізувати маршрути та адаптуватися до змін умов виконання місій.

2.4.2 Методи рою частинок

Метод рою частинок — це алгоритм оптимізації, натхненний поведінкою природних роїв, таких як зграї птахів або рої комах. Він забезпечує швидкий пошук рішень для задач оптимізації, особливо у випадках, пов'язаних з розподілом завдань та пошуком оптимальних маршрутів для БПЛА. Забезпечує швидкий пошук рішень

для оптимізації маршрутів та розподілу завдань.

До основних принципів методу відноситься популяція частинок. Кожне можливе рішення представляється частинкою в багатовимірному просторі. Кожна частинка має позицію та швидкість, які визначають її поточне рішення та напрямок руху.

В ітеративному процесі частинки "рухаються" по простору рішень, керуючись власним досвідом, досвідом сусідів або всієї популяції. Для пошуку оптимуму частинки орієнтуються на найкращу позицію, знайдену ними самими (локальний оптимум), і найкращу позицію, знайдену всіма частинками (глобальний оптимум). Адаптація це коли частинки коригують свою швидкість і траєкторію в залежності від отриманого досвіду, що дозволяє досягати оптимальних рішень.

На етапі ініціалізації створюється популяція частинок із випадковими початковими позиціями та швидкостями. Визначається цільова функція, яку потрібно мінімізувати або максимізувати. Для оцінки придатності обчислюється значення цільової функції для кожної частинки, яке визначає її якість (придатність). Оновлення найкращих позицій відбувається для кожної частинки зберігається її найкраща позиція (локальний оптимум). Визначається найкраща позиція серед усіх частинок (глобальний оптимум).

Швидкість частинки оновлюється за формулою:

$$v_i^{t+1} = w \cdot v_i^t + c_1 \cdot r_1 \cdot (p_i - x_i^t) + c_2 \cdot r_2 \cdot (g - x_i^t), \quad (2.6)$$

де

v_i^t – швидкість частинки i на ітерації t ;

x_i^t – позиція частинки i на ітерації t ;

p_i – найкраща знайдена позиція частинки i .

g – глобально найкраща позиція;

w – інерційний коефіцієнт (контролює баланс між глобальним і локальним пошуком);

c_1, c_2 – коефіцієнти навчання (визначають вагу локального та глобального пошуку);

r_1, r_2 – випадкові числа в діапазоні $[0, 1]$.

Нова позиція частинки обчислюється як:

$$x_i^{t+1} = x_i^t + v_i^{t+1}.$$

Процес триває, доки не досягнуто задану кількість ітерацій або значення цільової функції не буде достатньо близьким до оптимального.

Метод рою частинок є потужним інструментом для вирішення задач оптимізації в командній роботі БПЛА. Його здатність швидко знаходити рішення, адаптуватися до динамічних умов і враховувати взаємодії між агентами робить його ідеальним для управління та планування в складних системах.

2.4.3 Методи машинного навчання

Машинне навчання (ML) – це підхід до обробки даних, що дозволяє системам автоматично навчатися та вдосконалювати свої дії на основі накопичених даних. У контексті прогнозування та адаптації завдань для безпілотних літальних апаратів (БПЛА), ML надає інструменти для аналізу великих обсягів даних, створення прогнозів і динамічного коригування рішень відповідно до змінних умов. Особливо глибоке навчання, для прогнозування та адаптації завдань на основі великих обсягів даних.

Ключові етапи застосування методу ML включають дані про виконані завдання, маршрути, погодні умови, витрати енергії, пріоритети завдань, обмеження тощо. На попередній обробці даних відбувається видалення шуму, заповнення пропущених значень, нормалізація та категоризація даних.

Алгоритми можуть бути класифікаційними, регресійними або кластеризаційними в залежності від цілей:

- Прогнозування виконання завдання (регресія).
- Пріоритизація завдань (класифікація).
- Групування завдань або зон (кластеризація).

Навчання виконується на великій вибірці історичних даних із використанням алгоритмів, таких як:

- Лінійна регресія.
- Дерева рішень.
- Глибокі нейронні мережі (DNN).
- Підтримуючі вектори (SVM).

Далі здійснюється перевірка точності прогнозів на основі тестової вибірки. Після навчання модель використовується в реальному часі для прогнозування та адаптації завдань.

Метод машинного навчання є потужним інструментом для створення адаптивних систем управління завданнями в командній роботі з БПЛА. Завдяки аналізу великих обсягів даних, прогнозам та динамічному коригуванню, такі системи забезпечують ефективність, адаптивність та оптимізацію ресурсів.

Існуючі методи забезпечують широкий вибір підходів до планування завдань у командній роботі з БПЛА. Вибір методу залежить від умов роботи, пріоритетів завдань, рівня автономності апаратів та доступних ресурсів. Гібридні методи, які поєднують централізоване планування та децентралізоване управління, вважаються перспективними, оскільки забезпечують баланс між ефективністю та стійкістю системи

3 МОДЕЛЬ КОМАНДНОЇ РОБОТИ ЛЮДИНИ І БПЛА

3.1 Принципи коактивного проєктування

Коактивне проєктування – це підхід до розробки інтерактивних систем, зокрема робототехнічних і автономних систем, де людина та машина тісно співпрацюють для досягнення спільних цілей. Основна ідея полягає в створенні таких систем, які забезпечують ефективну взаємодію, взаємодоповнення можливостей і адаптивність до змінних умов.

Система та користувач працюють разом як єдина команда, використовуючи сильні сторони кожного для досягнення оптимального результату. Процеси управління або функціонування системи можуть змінюватися залежно від дій чи потреб користувача. Замість заміщення людини, система допомагає їй, виконуючи завдання, які важко або неможливо виконати самостійно. І система, і користувач можуть навчатися під час виконання завдань, покращуючи ефективність і взаєморозуміння. Проєктування враховує людську фізіологію, психологію та когнітивні можливості для забезпечення комфортної та ефективної взаємодії.

Коактивне проєктування застосовується для розробки роботів-помічників, які працюють поряд із людьми, наприклад, у медицині, виробництві чи рятувальних операціях, для спільного управління автономним транспортом, де людина може втручатися у критичних ситуаціях; для використання безпілотників або інших систем, які співпрацюють з операторами для виконання стратегічних завдань, в якості інструментів, які допомагають хірургам під час операцій, підвищуючи точність і безпеку, як системи, які адаптують навчальний контент під потреби студента, допомагаючи йому засвоювати матеріал у своєму темпі.

При коактивному проєктуванні людина та система розглядаються як взаємодоповнюючі елементи, кожен із яких виконує свої функції. Система повинна бути зрозумілою для користувача, пояснювати свої дії та прийняті рішення. Вона може змінювати свої функції або підхід залежно від умов чи вказівок користувача.

Мета такої взаємодії — мінімізувати час і зусилля, необхідні для досягнення результату. Людина повинна легко використовувати систему без ризику для здоров'я чи помилок через складність інтерфейсу.

У традиційному підході автономні системи проєктуються як незалежні або замітники людини. У коактивному проєктуванні система працює разом із людиною, а не замість неї, підлаштовуючись до змін середовища та потреб користувача.

В якості прикладу можна уявити рятувальну операцію, де група дронів разом із рятувальниками шукає постраждалих. Люди визначають загальну стратегію та напрямок пошуку, тоді як дрони автономне обробляють зображення місцевості, надсилають дані та пропонують оптимальні рішення для пошуку, враховуючи погодні умови чи небезпеки.

Коактивне проєктування забезпечує ефективну співпрацю між людиною та машиною, що робить системи більш гнучкими, адаптивними та орієнтованими на користувача.

3.2 Структура для командної роботи людей та БПЛА

Структура для командної роботи людини і БПЛА складається з чотирьох модулів, а саме: модуля коактивного проєктування, модуля планування завдань на основі агентів, модуля навчання з глибоким підкріпленням і модуля вибору дій зі змішаною ініціативою, як показано на рисунку 3.1.

Модуль коактивного проєктування відповідає за аналіз відносин взаємозалежності між членами команди людини або БПЛА під час спільної діяльності. Перш за все, описується загальний робочий процес діяльності та з'ясовуються необхідні здібності для діяльності. Потім вимірюються здібності кожного члена команди і відповідно до них призначається роль.

Як і в оригінальному підході коактивного дизайну, запропонованому Джонсоном, для забезпечення розуміння намірів і дій між учасниками команди необхідно дотримуватися принципів спостережливості, передбачуваності та керованості (OPD).

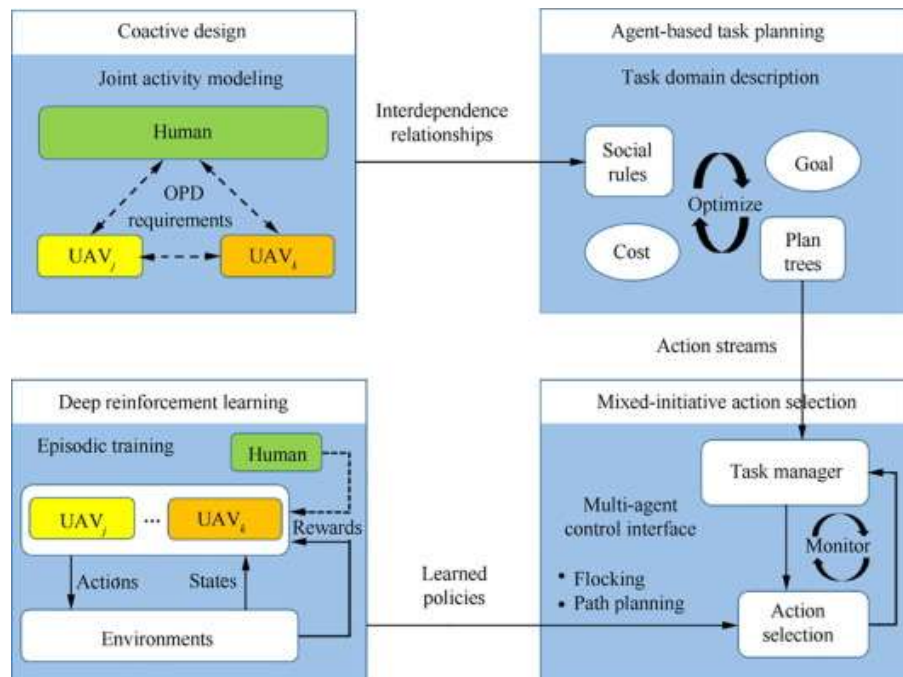


Рисунок 3.1 – Структура для прозорї командної роботи людей та БПЛА

Принципи спостережливості, передбачуваності та керованості (OPD — Observability, Predictability, Directability) є ключовими характеристиками взаємодії в складних системах, особливо тих, які включають людей та автономні системи, наприклад, у командній роботі з БПЛА. Ці принципи забезпечують ефективну співпрацю між учасниками системи, незалежно від їхнього типу (люди чи машини).

Спостережливість полягає в тому, що всі учасники системи (люди та машини) можуть отримувати необхідну інформацію про стан інших компонентів або агентів. Забезпечує прозорість дій і стану системи. Наприклад, в контексті БПЛА це означає, що кожен апарат у команді повинен "знати", що роблять інші апарати, та бути здатним ділитися своїм станом (наприклад, місцезнаходження, статус виконання задачі, рівень заряду).

Передбачуваність відображає можливість прогнозувати дії та наміри інших учасників на основі отриманих даних. У системах з автономними агентами це означає, що дії кожного агента мають бути зрозумілими і передбачуваними для інших. Наприклад, у кооперативній роботі БПЛА поведінка одного дрона не повинна суперечити логіці дій всієї команди.

Керованість – це здатність впливати на поведінку системи або окремих її компонентів для досягнення спільних цілей. Людина-оператор або центральний вузол управління повинні мати можливість змінювати або адаптувати плани дій агентів у реальному часі. У прикладі з БПЛА це означає, що оператор може вносити корективи в маршрути дронів або змінювати їхні завдання в залежності від ситуації.

На відміну від роботи Джонсона, в кваліфікаційній роботі розглядається значно більша команда, яка включає людей і різномірні БПЛА, що створює значно складніші взаємозалежності. У результаті, як ілюстрував Джонсон, ручне планування та декомпозиція завдань стають непридатними для нашого сценарію. Тому виникає потреба у впровадженні автоматизованого планування.

Всі відносини взаємозалежності фіксуються модулем планування завдань на основі агентів як частина соціальних правил. Інші соціальні правила включають зменшення часу очікування агента, збалансування кількості дій між агентами, зменшення зв'язку між потоками дій агентів, видалення деяких послідовностей небажаних дій та ін. Ці соціальні правила корисні для мультиагентних систем (MAS) для обчислення та обрізання дерев планів. Система планування на основі агентів чітко визначає мету завдання разом з відповідними функціями витрат. Результати дерев планів і потоків дій відображаються у графічному формалізмі, щоб показати, як завдання декомпонується, а також часові рамки дій кожного агента. Таким чином, результати планування стають зрозумілими для людини. У цій магістерській роботі пропонується використовувати ієрархічний агентний планувальник завдань (ІАПЗ) для вирішення проблеми планування у зрозумілій для людини спосіб.

У модулі глибокого навчання з підкріпленням (Deep Reinforcement Learning, DRL) ми навчили поведінці зграї та планувальника маршруту, щоб допомогти людині-оператору керувати пілотованим літальним апаратом БПЛА. По-перше, поведінка згуртування дозволяє загону БПЛА автономне згуртовуватися з лідером, тобто БПЛА, в безперервному просторі стану і дій. Зазначимо, що БПЛА може часто змінювати напрямок і швидкість польоту відповідно до вимог завдання, які, як правило, невідомі БПЛА в умовах обмеженого зв'язку. Таким чином, людина-оператор може зрозуміти поведінку БПЛА в режимі «лідер-послідовник». З іншого

боку, агент планування траєкторії навчається вибирати оптимальну політику дій для уникнення потенційних загроз під час навігації до заданого пункту призначення. Ці політики також можуть бути використані для надання пропозицій людині або сигналізації людині, коли MAV наближається до небезпечних зон. Таким чином, вивчені політики стають зрозумілими і контрольованими для прозорої командної роботи людини і БПЛА.

У модулі вибору дій зі змішаною ініціативою людина-оператор контролює виконання планів за допомогою багатоагентного інтерфейсу управління, який також відповідає вимогам OPD. Навчена поведінка зграї і планувальник шляхів також можуть бути активовані людиною-оператором через інтерфейс управління. Іншими словами, людина може спостерігати, прогнозувати і керувати поведінкою БПЛА за потреби. У цій магістерській роботі досліджується складне завдання в динамічному середовищі, що характеризується непередбачуваними подіями, такими як атака противника, порушення зв'язку або пошкодження безпілотних літальних апаратів.

3.3 Спільне проектування командної роботи людини та БПЛА

3.3.1 Опис завдання

Подібно до HADES (Hoplological Autonomous Defence and Engage Simulation), середовище командної роботи людини і БПЛА (HUT) показано на рис. 3.2.

Середовище HUT (Hierarchical Utility Tree) — це метод або інструмент, який використовується для ієрархічного моделювання цілей, задач або рішень у вигляді дерева. Його основна мета — структурувати проблеми чи сценарії, забезпечуючи ясність і полегшуючи процес прийняття рішень.

Основні характеристики середовища HUT:

- Всі цілі або задачі представлені у вигляді дерева, де вузли відображають підзадачі, а кінцеві гілки — конкретні рішення або критерії.
- Кожен вузол чи рішення оцінюється за певними показниками (утилітами), що дозволяє вибирати оптимальні варіанти.

– Може бути адаптовано для роботи з різними типами задач, наприклад, розподіл ресурсів, планування завдань, прийняття рішень у складних системах.

– Великі складні проблеми розбиваються на менші частини, що полегшує аналіз і управління.

HUT підходить для сценаріїв, де потрібно враховувати багато факторів та зберігати баланс між різними аспектами завдання або системи.

Наші середовища для командної взаємодії людини і БПЛА мають подібність до популярних стратегічних ігор у реальному часі, таких як StarCraft I і II, які часто використовуються як еталонні платформи для тестування сучасних алгоритмів мультиагентної координації. Типове середовище HUT включає дві команди агентів, які змагаються одна з одною.

Одна команда знаходиться під керівництвом людини-оператора, яка управляє трьома типами агентів:

Пілотованим літальним апаратом (ПЛА), оснащеним системами виявлення та озброєння (багатозадачний ПЛА).

БПЛА_З, який має лише системи виявлення (однозадачний апарат).

БПЛА_ЗД, обладнаним як системами виявлення, так і озброєнням (багатозадачний апарат).

Кількість та типи встановлених датчиків або ракет можуть варіюватися для кожного з цих апаратів, що визначає їх функціональні можливості.

Протилежна команда складається з кількох об'єктів, таких як критичні цілі, зенітно-ракетні комплекси (ЗРК), радіолокаційні станції тощо. Завдання команди людини полягає у знищенні критичних цілей з мінімальними втратами серед своїх БПЛА. Якщо будь-який БПЛА буде знищено силами противника, це може призвести до провалу місії.

Також середовище HUT дозволяє задавати різні конфігурації складів та розташування для обох команд, забезпечуючи широкий спектр сценаріїв для моделювання.

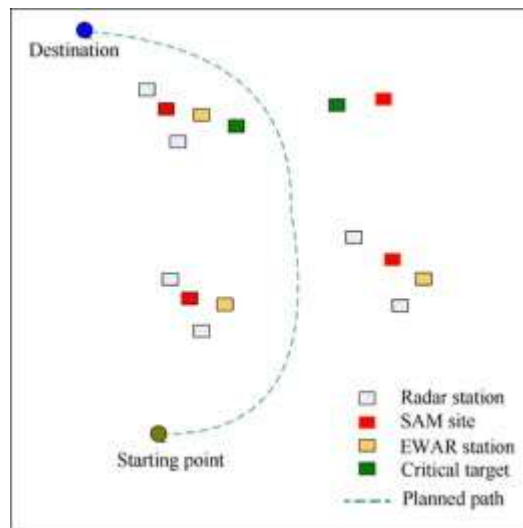


Рисунок 3.2 – Середовище спільної роботи людини та БПЛА (HUT)

MAV управляється вручну людиною-оператором. MAV (пілотований літальний апарат) управляється вручну людиною-оператором, що означає, що всі його дії безпосередньо контролюються людиною в реальному часі. Це включає такі аспекти, як навігація, ухвалення рішень щодо виконання завдань, використання сенсорів для збору даних, а також застосування систем озброєння, якщо такі є. На відміну від автономних безпілотних апаратів, MAV не діє самостійно, а повністю залежить від команди оператора. Це дозволяє людині враховувати змінні умови в реальному часі, але також вимагає високої уваги, швидкої реакції та глибоких знань від оператора.

БПЛА може приймати обґрунтовані або необґрунтовані рішення щодо вибору дій. Потім оператор може спостерігати з інтерфейсу управління, як БПЛА реагують на рішення MAV, наприклад, попереджають оператора про потенційну небезпеку або пропонують кращий шлях для вирішення завдання. При цьому оператор може прийняти або накладити вето на рішення БПЛА. Таким чином, вибір дій БПЛА є змішано-ініціативним, а співпраця стає зрозумілою.

3.4 Аналіз взаємозалежності

Аналіз взаємозалежності (АЗ) з використанням принципів коактивного проектування. У середовищі HUT виконання завдань зазвичай поділяється на кілька основних етапів: планування, розвідку, атаку та оцінку. Кожен із цих етапів ділиться

на окремі виконавчі підзадачі. Для кожної підзадачі визначаються ролі виконавця або помічника, залежно від необхідних можливостей і реальних здібностей члена команди.

Система кольорової індикації позначає можливості виконання підзадачі:

- Зелений — завдання може бути виконане самостійно.
- Жовтий — виконавець потребує підтримки.
- Помаранчевий — помічник може надати необхідну допомогу.
- Червоний — відсутня можливість виконати підзадачу.

Крім того, для кожної підзадачі аналізуються вимоги до ОПР (спостережливості, передбачуваності, керованості).

Важливо враховувати, що члени команди мають різні можливості. Наприклад, MAV здатний нести великі капсули РЕБ (радіоелектронної боротьби) і протирадіаційні ракети, що недоступно для БПЛА. Водночас БПЛА мають перевагу у розвідці завдяки тривалому перебуванню в цільових зонах, навіть якщо вони небезпечні для інших апаратів. Однак БПЛА мають обмежені можливості у випадках надзвичайних подій або загроз. У таких ситуаціях MAV може надати підтримку у прийнятті рішень.

3.5 Алгоритм взаємодії агентів

Агенти здатні ефективно розподіляти ресурси для оптимального виконання завдань пошуку та рятування [24]. Завдяки загальній стратегії кожен агент виконує свою функцію, що сприяє швидшому та ефективнішому виявленню постраждалих і вирішенню небезпечних ситуацій. Динамічність умов у небезпечному середовищі вимагає оперативного коригування стратегій та тактик. Координація агентів забезпечує гнучкість у адаптації до нових обставин і змін у планах дій. Міжагентне спілкування та взаємодія з центральним командуванням дозволяють оперативно обмінюватися важливою інформацією й узгоджувати дії для досягнення спільної мети.

Кожен агент є автономною рятувальною одиницею, оснащеною

різноманітними сенсорами (тепловими, відео, звуковими тощо) і засобами комунікації для взаємодії з іншими агентами. Робочим середовищем виступає велика територія, що охоплює ліси, гори та водойми, де складні умови навігації можуть ускладнювати виконання завдань. Основними ризиками середовища є небезпечні тварини, несприятливі погодні умови, зсуви, обвали та пожежі.

Мета сценарію є максимально швидке знаходження та рятування зниклих або постраждалих осіб із мінімізацією ризиків для рятувальників.

Сценарій координації агентів у завданні пошуку та рятування в небезпечному середовищі включає наступні етапи:

1. Аналіз середовища. Агенти починають із збору та аналізу даних про місцевість, погодні умови, можливі небезпеки та інформацію про постраждалих.

2. Розподіл завдань. На основі зібраних даних завдання розподіляються між агентами залежно від їхніх функціональних можливостей. Наприклад, один агент може відповідати за організацію пошуку, а інший – за обробку інформації про постраждалих.

3. Взаємодія та координація. Агенти обмінюються інформацією між собою та з центральним командуванням. Наприклад, агент, який виявив постраждалого, передає його координати для швидкого реагування інших.

4. Планування маршрутів. Агенти колективно складають оптимальні маршрути для пошуку та рятування, враховуючи ризики й потенційні перешкоди.

4. Адаптація до змін. У разі змін у середовищі агенти оперативно коригують стратегії та плани дій, щоб адекватно реагувати на нові виклики.

5. Оцінка результатів. Після успішного виконання місії агенти аналізують результати операції, обговорюють ефективність дій та проводять завершальні етапи місії.

Такий підхід дозволяє агентам злагоджено працювати, досягаючи спільної мети – рятування людських життів і збереження майна в екстремальних умовах.

4 ПЛАНУВАННЯ ЗАВДАНЬ НА БАЗІ АГЕНТІВ

4.1 Ієрархічний планувальник

Розглянемо ієрархічний планувальник завдань на основі агентів – підхід до планування, який організовує процес розподілу і виконання завдань у багаторівневій структурі. У цій системі агенти (віртуальні або фізичні, наприклад, дрони чи програмні модулі) виконують завдання, координуючись між собою за допомогою ієрархічної структури управління. Такий планувальник дозволяє ефективно розподіляти ресурси, адаптуватися до змін і вирішувати складні проблеми в динамічному середовищі.

Ієрархічний планувальник складається з трьох рівнів:

Верхній рівень (стратегічний) формує загальну мету і визначає основні етапи виконання завдань. Цей рівень здійснює глобальне планування та розподіл завдань між групами агентів.

Середній рівень (тактичний) деталізує завдання, отримані від стратегічного рівня, та розподіляє їх між окремими агентами або підгрупами.

Нижній рівень (оперативний) виконує конкретні завдання, використовуючи ресурси та враховуючи локальні обмеження.

У ієрархічному планувальнику завдань рівні взаємодіють через передачу інформації, завдань і рішень між стратегічним, тактичним і оперативним рівнями. Ця взаємодія забезпечує координацію агентів, адаптацію до змін у середовищі та ефективне виконання завдань.

Передача завдань здійснюється зверху вниз (декомпозиція завдань). Стратегічний рівень формує загальну стратегію та високорівневі цілі, наприклад: "Забезпечити контроль над визначеною територією."

Цілі передаються на тактичний рівень, де вони розбиваються на менші підзадачі. Наприклад: "*виконати розвідку місцевості*", "*знешкодити ворожі об'єкти*."

Оперативний рівень отримує ці підзадачі і деталізує їх до конкретних інструкцій, наприклад:

"Дрон №3 виконує обліт території на висоті 200 м."

"Дрон №5 атакує об'єкт за координатами X, Y."

Зворотний зв'язок здійснюється передачею інформації знизу вгору. Оперативний рівень постійно передає дані про стан виконання завдань та виявлені проблеми чи зміни в середовищі.

Тактичний рівень аналізує отриману інформацію, коригує поточні дії та передає узагальнені дані стратегічному рівню.

Стратегічний рівень переглядає загальну стратегію на основі отриманих даних і, за необхідності, змінює пріоритети.

Адаптація до змін відбувається таким чином. Якщо на оперативному рівні виявлено нові загрози або виникли труднощі у виконанні завдання, інформація передається вгору.

Тактичний рівень перерозподіляє ресурси або змінює розподіл ролей між агентами.

Стратегічний рівень може змінити пріоритети завдань або навіть визначити нову мету для всієї системи.

Для взаємної підтримки рівні діють як взаємозалежні. Верхні рівні забезпечують узгодженість дій всієї системи та враховують глобальні фактори. Нижні рівні фокусуються на деталях виконання завдань, пропонуючи дані для вдосконалення загальної стратегії.

Наприклад. Стратегічний рівень визначає: "Контролювати регіон R."

Тактичний рівень ділить завдання:

– *"Виконати розвідку регіону R."*

– *"Знешкодити вороже ППО в секторі S."*

Оперативний рівень:

Дрон А: *"Виконує обліт по маршруту M."*

Дрон В: *"Атакує ціль за координатами X, Y."*

Якщо дрон В зазнає пошкоджень, інформація передається нагору. Тактичний

рівень перепризначає атаку іншому дрону, а стратегічний переглядає цілі, щоб уникнути великих втрат.

Ієрархічна структура дозволяє масштабувати планувальник, зберігаючи чіткість у стратегічних цілях та ефективність у виконанні завдань.

Таким чином, кожен агент виконує свою частину роботи відповідно до свого рівня в ієрархії.

Агенти взаємодіють один з одним, обмінюючись інформацією про стан виконання завдань, ресурси і зміни в середовищі.

Завдання спочатку розбиваються на підзадачі на вищих рівнях.

Підзадачі передаються до нижчих рівнів, де вони додатково уточнюються або виконуються.

Планувальник постійно адаптує дії агентів до змін у середовищі, таких як поява нових задач, відмова агентів чи змінення пріоритетів.

Ієрархічний планувальник завдань на основі агентів забезпечує високу ефективність і гнучкість у складних системах, дозволяючи агентам діяти як автономно, так і в рамках спільної стратегії.

Ієрархічний планувальник завдань на основі агентів (ІАПЗ) може вирішити проблему планування завдань у спосіб, зрозумілий людині. ІА надає корисні взаємозв'язки для ІАПЗ, щоб генерувати дерева планів і потоки дій, а оптимальність запланованого рішення гарантується глобальними обмеженнями соціальних правил і визначеними користувачем функціями витрат.

4.2 Опис предметної області

Опис предметної області передбачає структуроване представлення об'єктів, їхніх властивостей, а також взаємозв'язків між ними. Пропонується це зробити наступним чином.

Для визначення сутностей (об'єктів) спочатку ідентифікуємо всі об'єкти, які беруть участь у виконанні задачі. Кожна сутність має унікальне ім'я, яке дозволяє

однозначно її ідентифікувати. Наприклад: агент (людина чи БПЛА), ціль (об'єкт для виконання задачі), середовище (локація, умови).

Атрибути визначені як статичні та динамічні. Статичними атрибутами будемо вважати властивості, які залишаються незмінними впродовж усієї задачі, наприклад, для БПЛА: максимальна вантажопідйомність, швидкість; для цілі: місцезнаходження, розмір. Динамічними атрибутами будемо вважати властивості, які змінюються під час виконання задачі, наприклад, для БПЛА: поточний заряд батареї, координати; для цілі: стан (активна, знешкоджена).

Опис взаємозв'язків між сутностями відображають залежності або ієрархії. Наприклад, БПЛА закріплений за певною місією.

Місія складається з кількох завдань, пов'язаних із конкретними цілями.

Для формалізації дій визначимо дії, які можуть виконуватися в межах задачі.

Кожна дія повинна мати:

- вхідні дані (параметри, на які впливає дія);
- умови виконання (обмеження чи передумови);
- результат (зміни в атрибутах чи середовищі).

Наприклад, дія "*атакувати ціль*" вимагає доступного заряду та знаходження в радіусі дії.

Визначенням ієрархії задач є розподіл задач на підзадачі для більш детального управління. Наприклад, головна задача це захоплення об'єкта; підзадачі – розвідка місцевості, нейтралізація загроз, доставка вантажу.

Також необхідно визначите обмеження, які впливають на виконання задач (наприклад, кількість ресурсів, зона покриття, час виконання).

Формалізація цілей полягає в описі основної мети системи та критеріїв її досягнення. Наприклад, знищення всіх критичних цілей із мінімальними втратами.

Такий підхід забезпечує чітке, зрозуміле та формалізоване уявлення про предметну область, що є основою для побудови планувальників або інших систем керування.

Будемо використовувати об'єктно-орієнтовану парадигму програмування, де кожен елемент у проблемній області представлений як об'єкт або сутність. Агент є

сутністю першого класу, здатною виконувати послідовність дій. Хоча агенти визначаються заздалегідь, їхні атрибути можуть приймати різні значення. Кожна сутність має унікальне ім'я та набір атрибутів, які визначають зв'язки між сутностями. Атрибути поділяються на статичні та динамічні.

Динамічні атрибути можуть змінюватися під час розв'язання задачі. Статичні атрибути задаються лише під час ініціалізації й не змінюються надалі.

Атрибути також можуть бути атомарними або множинними: атомарний атрибут має одне значення; множинний атрибут може мати кілька значень..

Приклад визначення сутності наведено у лістингу 4.1.

Лістинг 4.1 – Приклад визначення сутності:

```
// entity types
define entityType Location;
define entityType Formation;
// Entity attributes
define entityAttributes Agent {
  //Agent is predefined
  static atom string type;
  dynamic atom Location at;
  dynamic atom bool isAssigned;
  dynamic atom bool isRtPlanned;
  dynamic set Location path; }
define entityAttributes Formation {
  Static set Agent member;}
```

Цей код написаний мовою GAML (GAMA Modeling Language), яка використовується в середовищі моделювання GAMA для створення багатофункціональних агентно-орієнтованих моделей. Ця мова застосовується для розробки агентно-орієнтованих моделей у різних галузях, таких як екологія, соціальні науки, логістика тощо.

Після того, як сутність визначено, її потрібно конкретизувати, а її атрибутам присвоїти значення. Початковий стан задачі визначається початковими атрибутами кожної сутності. Опис початкових станів повинен бути максимально повним і детальним, щоб уникнути двозначності. Приклад оголошення сутності та присвоєння атрибутів наведено у лістингу 4.2. У цьому прикладі символ <<== вказує на додавання елемента до набору атрибутів.

Лістинг 4.2 – Приклад ініціалізації станів:

```

/Entity declaration
MAV=new Agent;
UAVz1=new Agent;
Formation1=new Formation;
//Agent state initialization
MAV.type = "MAV";
MAV.at = AP;
UAVz1.type = "UAVz";
UAVz1.at = AP;
UAVz1.isAssigned=false;
UAVz1.isRtPlanned=false;
//Formation sate initialization
Formation1.member<<=UAVz1;

```

4.3 Декомпозиція завдання

Метою планування є перетворення загальної, складної задачі (високорівневої задачі) на низку конкретних, простих дій (атомарних задач), які можуть бути виконані системою або агентами. Спочатку визначимо, що таке високорівнева та атомарна задачі. Високорівнева задача – це завдання, яке задається на рівні абстракції, що не містить конкретних інструкцій або дій. Наприклад, "*забезпечити безпеку об'єкта*" або "*доставити вантаж до пункту призначення*". Атомарні задачі – це найменші,

неподільні дії, які система або агент може виконати, наприклад, "переміститися в точку X", "сканувати зону Y", "запустити дрон Z".

Планувальник розбиває високорівневу задачу на підзадачі, які, своєю чергою, можуть бути розбиті на ще менші задачі. Процес триває доти, доки всі задачі не будуть перетворені на атомарні, тобто такі, які система може виконати без додаткового розділення:

1. Визначаємо високорівневу задачу: "розвідати місцевість".
2. Робимо декомпозицію – переміститися до зони спостереження.
3. Скануємо територію.
4. Аналізуємо зібрані дані.
5. Робимо атомарні задачі:
 - Запускаємо двигун.
 - Рухаємося на північ із заданою швидкістю.
 - Виконуємо сканування на радіус 100 м.

Такий підхід забезпечує структуроване виконання задачі, полегшує розподіл завдань між агентами та дозволяє реагувати на зміни у середовищі, коригуючи лише конкретні дії. Отже, навіть складні задачі можуть бути вирішені поетапно, що є ключовим для автоматизованих систем та багатозадачних середовищ.

Приклад визначення методу ReconMission наведено у лістингу 4.3. У цьому прикладі символ \gg позначає судження про приналежність елемента до множини атрибутів. Метод ReconMission стосується створення плану для виконання розвідувальної операції з урахуванням агентів, їхніх ролей і послідовності підзадач.

Лістинг 4.3 –Визначення методу ReconMission:

```
method Recon(Formation F, Location L){
  goal{L.isReconned==true;};
  {
    preconditions{};
    subtasks{
      M=SELECT(Agent, {M.type == "MAV";});
```

```

UZ = SELECT (Agent, {UZ >> F.member; UZ.type == "UAVz";});
1:Search (UZ, L);
2:Locate (M, L) > 1;
3:Surveillance (UZ, L) > 2;};}

```

Дія - це найнижчий рівень декомпозиції завдання. Структура дії включає набір передумов, набір наслідків, функцію вартості та функцію тривалості. Surveillance — визначає дію спостереження, яка виконується для певної локації за допомогою агента: Agent UZ – агент, який виконує дію (обов’язково має бути типу "UAVz"); Location TL – локація, де відбувається спостереження. Код у лістингу 4.4 описує дію Surveillance (спостереження), яка виконується агентом UZ (типу "UAVz") у визначеній локації TL.

Лістинг 4.4 – Приклад визначення дії:

```

action Surveillance (Agent UZ, Location TL) {
  preconditions {
    UZ.type == "UAVz";
    TL.isReconned == false;
    TL.isLocated == true;
  };
  effects {
    TL.isReconned = true;
  };
  cost {costFn (1) };
  duration {durationFn (5) };
}

```

Передумови описують контекст, в якому ця дія може бути виконана: UZ.type == "UAVz" – агент, який виконує дію, повинен бути типу "UAVz"; TL.isReconned == false – локація ще не була розвідана; TL.isLocated == true – локація вже визначена. Ці умови гарантують, що дія виконується лише тоді, коли відповідний агент доступний і локація готова для розвідувального спостереження. Ефекти представляють очікувані

результати цієї дії: `TL.isReconned = true` – після завершення дії локація стає позначеною як розвідана.

Це означає, що виконання дії оновлює стан середовища, позначаючи, що розвідка завершена. Функція витрат відображає вартість виконання цієї дії, включаючи економічні витрати або витрати часу: `cost{costFn(1)}` — вартість виконання дії визначається функцією `costFn` із переданим значенням 1. Це може представляти, наприклад, ресурсні витрати (пальне, енергія тощо). Функція тривалості вказує на період часу, необхідний для завершення дії, який також використовується для координації часових рамок між потоками дій декількох агентів: `duration{durationFn(5)}` — тривалість виконання дії визначається функцією `durationFn` із переданим значенням 5. Це може бути час, необхідний для завершення спостереження в одиницях, прийнятих у системі. Основна мета дії – забезпечити виконання спостереження, після чого локація TL буде позначена як розвідана (`isReconned == true`), що є важливим для досягнення більш глобальних завдань планування.

4.4 Формування плану

Планувальник ІАПЗ спочатку виконує ітерації декомпозиції на основі визначених методів, а потім уточнює проблему у вигляді дерева планів. Основною метою оптимізації ІАПЗ є контроль витрат. Після розрахунку дерева планів отримані підзадачі призначаються кожному агенту для формування потоку дій. Важливою перевагою ІАПЗ є те, що для злагодженої командної роботи можна визначити соціальні правила. Наприклад, можна скоротити час очікування, щоб зробити потоки дій різних агентів якомога компактнішими. Балансування зусиль також корисне для збалансування кількості дій між агентами. Небажаних послідовностей дій можна уникнути, якщо людина не хоче, щоб вони відбувалися. Крім того, складність можна контролювати, щоб зменшити зв'язок між потоками дій агентів, тобто зменшити взаємозалежності між агентами. Іншими словами, перераховані вище характеристики

роблять ІАПЗ хорошим вибором зрозумілого планувальника для командної роботи людини і БПЛА.

4.4.1 Глибоке навчання стадної поведінки з глибоким підкріпленням

Глибокі нейронні мережі є потужними апроксиматорами функцій, які можуть узагальнювати багатовимірні або неперервні простори станів і дій. Таким чином, агенти можуть мати справу з новими ситуаціями, з якими вони ніколи не стикалися раніше. У нашому сценарії ми припускаємо, що є один лідер і кілька послідовників, які летять з постійною середньою швидкістю і на різних висотах, щоб спростити проблему зіткнень. Оскільки послідовники не зіштовхуються, вони можуть використовувати спільну політику управління. Поведінку лідера визначає його оператор. Він ділиться інформацією про свій стан з послідовниками через бездротовий канал зв'язку, включаючи своє місцезнаходження і кут крену. Послідовник повинен контролювати кут крену і швидкість, щоб підтримувати певну відстань від лідера (рис. 4.1).

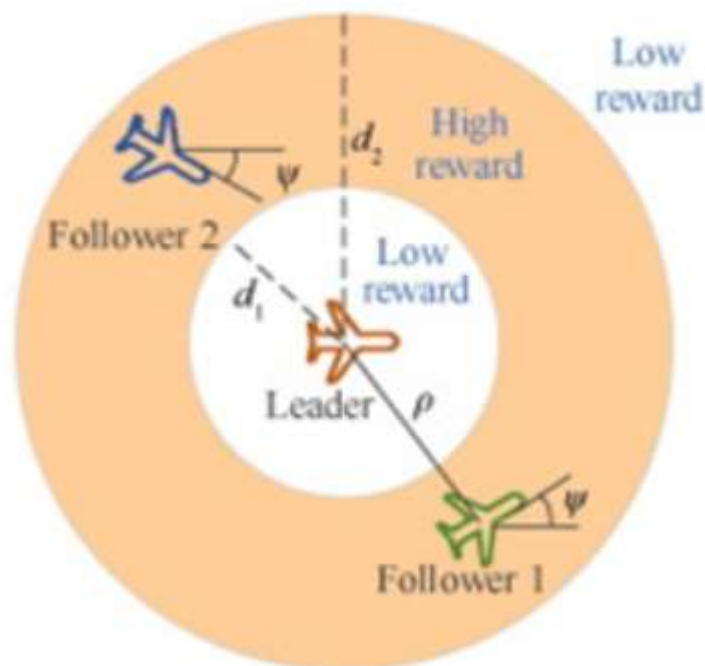


Рисунок 4.1– Вид зверху на топологію лідер послідовник

Розглянемо проблему «лідер-послідовник» як Марківський процес прийняття рішень (MDP), що передбачає формалізацію взаємодії між агентами (лідером і послідовниками) в контексті задачі, яку вони спільно виконують.

Основними елементами MDP є стан середовища (State, S), дії (Action, A), модель переходу станів (Transition Model, P), винагорода (Reward, R), політика (Policy, π), горизонт планування.

Стан середовища (State, S), що описує всю необхідну інформацію про поточну ситуацію в системі. У задачі «лідер-послідовник» стан може включати позицію лідера; позиції та орієнтації послідовників; відносну відстань і напрямок між лідером і кожним послідовником; поточний етап завдання (наприклад, початок, виконання, завершення); перешкоди та конфігурацію середовища. Формально S_t — стан середовища на час t .

Дії визначають, що можуть робити агенти в середовищі. Лідер визначає напрямок руху, швидкість або координати цільового стану. Послідовники змінюють свою позицію чи орієнтацію відповідно до команд лідера або стратегій слідування. Дії лідера та послідовників можуть бути визначені як вектор A_t , що включає індивідуальні дії кожного агента на час t .

Модель переходу описує ймовірність переходу зі стану S_t у стан S_{t+1} під дією A_t :

$$P(S_{t+1} | S_t, A_t)$$

У нашій задачі «лідер-послідовник» це залежить від: динаміки руху агентів, поведінкових правил слідування (наприклад, «зберігати певну відстань від лідера»), перешкод у середовищі.

Політика визначає, які дії агенти повинні виконувати в кожному стані, щоб максимізувати очікувану винагороду. Політика лідера $\pi_L(S_t)$ визначає, куди рухатися; політика послідовників: $\pi_F(S_t)$ визначає, як реагувати на дії лідера.

Горизонт може бути кінцевим, тобто завдання має визначений час завершення або кількість кроків, або нескінченним – завдання без чіткої кінцевої точки (наприклад, постійний патруль).

Тоді формалізація Марківського процесу прийняття рішень буде такою:

$$MDP = (S, A, P, R, \gamma)$$

де S – множина станів;

A – множина дій;

$P(S_{t+1} | S_t, A_t)$ – модель переходу станів;

$R(S_t, A_t)$ – функція винагороди;

γ – коефіцієнт дисконтування, що враховує важливість майбутніх винагород.

Завданням є знаходження оптимальної політики π_L і π_F для лідера та послідовників, які максимізують очікувану суму винагород:

$$\max_{\pi} E \left[\sum_{t=0}^{\infty} \gamma^t R(S_t, A_t) \right]$$

При такому підході можливі проблеми взаємодії між агентами (лідером і послідовниками), які формують багатокomпонентну MDP або кооперативну гру. Також агенти можуть мати обмежене сприйняття середовища або станів інших агентів, а середовище може змінюватися через рух перешкод або зміну мети.

Структура мережі алгоритму флокінгу показана на рисунку 4.4. Алгоритм флокінгу — це метод керування багатьма агентами (наприклад, роботами, дронами чи віртуальними істотами), який імітує поведінку зграй у природі (зграї птахів, косяки риб тощо). Алгоритм дозволяє групі агентів рухатися як єдиний організм, забезпечуючи збереження формації, уникнення зіткнень і адаптацію до змін у середовищі. Алгоритм базується на наборі простих локальних правил, які виконуються кожним агентом. Зазвичай використовується три основні правила, запропоновані Крейгом Рейнольдсом у 1986 році для симуляції поведінки зграї («Voids»). Кожен агент намагається вирівняти свій напрямок руху із середнім напрямком сусідніх агентів. Метою є збереження узгодженості в русі. Агент прагне до центру мас своїх сусідів з метою утримання зграї разом. Агент намагається триматися на безпечній відстані від сусідів, щоб уникнути зіткнень.

Розмірність представлення станів розширено з 6 до 9 параметрів. Як актор, так і критик мають схожу архітектуру багатошарового перцептрона (MLP), що складається з трьох повно з'єднаних шарів із 512, 512 і 256 нейронами відповідно. Після кожного щільного шару додається функція активації ReLU (випрямлена лінійна одиниця). Після трьох щільних шарів актор обробляє вхідний стан для обчислення крену та команд швидкості для БПЛА. Для забезпечення, що вихід актора знаходиться в межах $[-1, +1]$, використовується гіперболічна тангенціальна функція активації (tanh) на вихідному шарі. Оскільки простір дії для крену складає $[-10, +10]$ $[-10, +10]$, вихід актора масштабують шляхом множення на константу 10. Критик також використовує три щільні шари для оцінки значення стану, але його вихідний шар застосовує лінійну функцію активації для отримання результату.

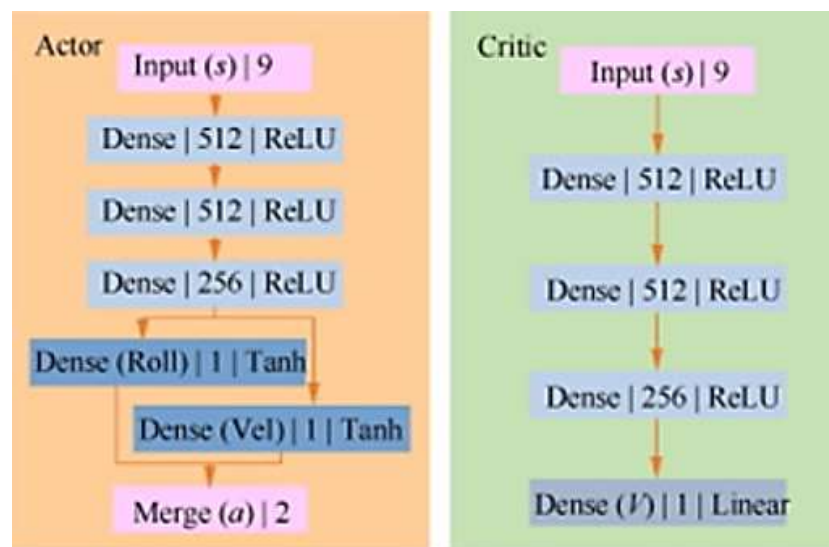


Рисунок 4.2 – Структура мережі алгоритму флокінгу

Зокрема, використовується багатошаровий перцептрон (MLP) для представлення актора, який відображає простір станів у простір дій, тобто, $Act^* : S \rightarrow A$, де $Act^*(s)$ позначає оптимальну дію в стані s . Ми також використовуємо МЛП для представлення критерію, який апроксимує функцію цінності стану: $V : S \rightarrow \mathbf{R}$. Позначимо через: (s_k, a_k, r_k, s_{k+1}) кортеж стану, дії та винагороди на кроці k та $k + 1$ відповідно. Правило оновлення критика визначається наступним чином:

$$V_{k+1}(s_k) = V_k(s_k) + \beta \delta^k$$

в якому TD-помилка δ^k це:

$$\delta^k = r_k + \gamma * V_k(s_{k+1}) - V_k(s_k)$$

де $0 \leq \gamma \leq 1$ коефіцієнт дисконтування, а $0 \leq \beta \leq 1$ це швидкість навчання критика.

На відміну від критика, актор оновлюється тільки тоді, коли TD-помилка позитивна, тобто поточний стан кращий за очікуваний. Таким чином, актор оновлюється наступним чином:

$$Act_{k+1}(s_k) = \begin{cases} Act_k(s_k) + \alpha(\alpha_k - Act_k(s_k)) & \text{if } \delta^k > 0 \\ Act_k(s_k) & \text{otherwise} \end{cases}$$

де $0 \leq \alpha \leq 1$ швидкість навчання актора.

Для вибору дослідницьких дій використовується гауссівське дослідження. Іншими словами, дія вибирається випадковим чином з гауссівського розподілу $G(x, \mu, \sigma)$ з центром на поточному виході актора $Act_k(s_k)$:

$$G(x, Act_k(s_k), \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-Act_k(s_k))^2/2\sigma^2}$$

де σ це гауссівський параметр розвідки.

Лідер контролюється оператором, тому ми розглядаємо лідера як частину середовища для послідовників. У кожному навчальному епізоді випадковим чином ініціалізуються як послідовник, так і лідер. Потім агент-послідовник отримує стан системи s_k , об'єднуючи свій власний стан і стан лідера. Агент обирає дію a_k відповідно до Гауссового пошуку на основі результатів роботи мережі акторів. Отже, кінематична модель БПЛА генерує наступний стан системи s_{k+1} , і дається негайна

винагорода r_k . Кожен кортеж (s_k, a_k, r_k, s_{k+1}) зберігається в пам'яті відтворення досвіду. Якщо пам'ять відтворення переповнена, найстаріші дані будуть замінені на найновіші. Оновлення параметрів мережі відбувається шляхом випадкової вибірки міні-партії з пам'яті відтворення.

4.4.2 Глибоке навчання з підкріпленням планувальника траєкторії БПЛА

Оцінка ситуації (SA) важлива через потенційні загрози з боку ворожих підрозділів. Модуль SA є зовнішньою функцією планувальника маршруту БПЛА (рис. 4.3). Модуль оцінки ситуації агента використовує отримані дані для побудови карти ситуації. Потім запропонований модуль D3QN використовує збережені карти ситуацій для прогнозування значень Q, що відповідають усім допустимим діям. Відповідно до прогнозованих значень Q, модуль вибору дій обирає дію, і номер обраної дії надсилається в навколишнє середовище. Нарешті, попередньо запланований шлях БПЛА буде оновлено, а потім БПЛА полетить на нову позицію, використовуючи обрану дію.

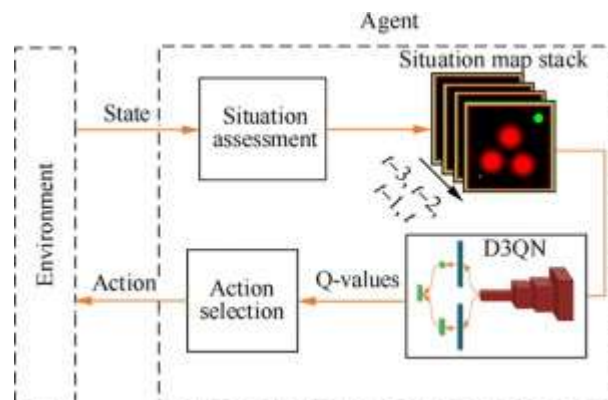


Рисунок 4.3 – Глибоке навчання з підкріпленням планувальника траєкторії БПЛА

Будується модель оцінки ситуації, використовуючи відстань між БПЛА та центром ворожої одиниці, наприклад, радаром раннього попередження або зенітно-ракетним комплексом (ЗРК). Зокрема, як тільки БПЛА буде виявлений радаром, він буде знищений зенітно-ракетним комплексом з певною ймовірністю, яка розраховується з урахуванням відстані між ними. Цей розрахунок враховує

максимальну дальність виявлення радаром, максимальний радіус зони ураження ракетою та максимальну дальність зони, з якої неможливо втекти. В результаті, рівень загрози кожної одиниці оборони для БПЛА розраховується наступним чином:

$$T_p = \begin{cases} 0 & D \geq R_{Rmax} \\ 0.5e^{-\frac{D-R_{Mmax}}{R_{Rmax}-R_{Mmax}}} & R_{Mmax} \leq D \leq R_{Rmax} \\ 2^{-\frac{D-R_{Mmax}}{R_{Rmax}-R_{Mmax}}} & R_{Mkmax} \leq D \leq R_{Mmax} \\ 1 & D < R_{Mkmax} \end{cases},$$

де D - відстань між БПЛА та ворожим підрозділом; R_{Rmax} - максимальна дальність виявлення радара; R_{Mmax} - максимальний радіус зони ураження ракетою; R_{Mkmax} - максимальна дальність зони ураження ракетою без можливості відхилення.

Рівень загрози T_p вказує на ймовірність ураження ЗРК. Ймовірності ураження різними ракетами вважаються незалежними. Таким чином, загальну загрозу для БПЛА від усіх ворожих одиниць можна розрахувати за наступним рівнянням:

$$Ts = 1 - \prod_{i=1}^k (1 - T_{pi}),$$

де k - кількість ворожих одиниць у зоні завдання БПЛА; T_{pi} - ймовірність ураження БПЛА при проходженні i -тої ворожої одиниці.

Для представлення результатів запропонованої моделі SA використовується колірна модель RGB, а потім відповідно будуються ситуаційні карти. За допомогою моделі оцінки ситуації для кожної позиції розраховується синтетичне значення загрози T_s , яке потім конвертується у значення пікселів кольорового каналу для побудови карти ситуації:

$$C = C_{min} + \frac{T_s - T_{min}}{T_{max} - T_{min}} (C_{max} - C_{min}),$$

де C_{max} і C_{min} позначають максимальне і мінімальне значення пікселів кольорового каналу відповідно; T_{max} і T_{min} - максимальне і мінімальне синтетичне значення загрози

на основі даних про ситуацію. У нашому випадку $C_{\max} = 255$ і $C_{\min} = 0$. Потім значення загрози конвертується у значення пікселів червоного каналу. Результуюча карта ситуації показана на рис. 4.4.

На відміну від моделі MDP, модель MDP побудована в дискретних просторах базується на карті ситуації. Три кути крену $\{-15, 0, 15\}$ витягуються з безперервного простору дії. Крім того, функція винагороди визначається наступним чином: якщо агент потрапляє в зону дії наземного радара, винагорода становить -2; якщо агент потрапляє в зону дії ЗРК, винагорода становить -4; якщо агент прибуває в заданий пункт призначення, винагорода становить 10.

Для того, щоб оцінити функцію вартості стану-дії (Q-функцію) БПЛА, створено дуельну подвійну глибоку Q-мережу (Dueling Double Deep Q-network, D3QN) як показано на рис. 4.4.

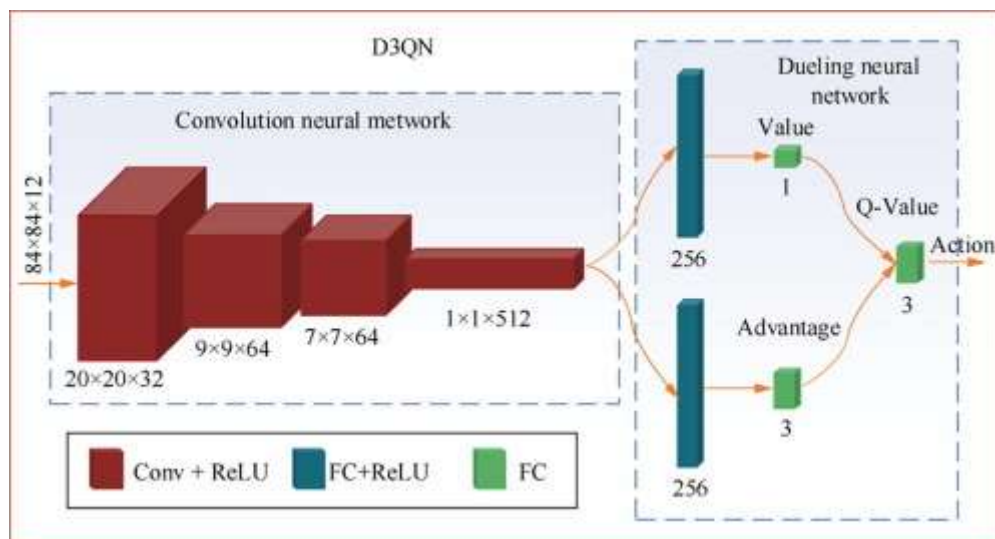


Рисунок 4.4 – Архітектура подвійної глибокої Q-мережі

На вхід мережі подається стек карт ситуації $84 \times 84 \times 12$, побудований за допомогою моделі оцінки ситуації. За ним слідує чотири згорткових шари та два повних з'єднаних шари для двох потоків дуельної архітектури. Зауважимо, що параметри мережі збігаються з параметрами попередньої роботи. Псевдокод запропонованого алгоритму показано в *Алгоритмі 1*.

Агенти БПЛА можуть використовувати ракети класу «повітря-земля» для знищення ворожих зенітно-ракетних комплексів, що знизить рівень загроз при наближенні до критичної цілі. Але використання ракет може також збільшити небезпеку бути знищеним підрозділами оборони противника. Тому агенти БПЛА повинні розумно розподіляти свої ракети відповідно до розподілу підрозділів противника. Іншими словами, агенти БПЛА можуть вибрати атаку ракетних підрозділів противника, щоб збільшити ймовірність виживання, або вони можуть вибрати атаку критичних цілей для вирішення поставленого завдання. У першому випадку зброї, що залишилася, може бути недостатньо для виконання завдання. У другому випадку ймовірність виживання може знизитися. Ця проблема прийняття рішень є складною, оскільки сценарій завдання є динамічним і не повністю відомим заздалегідь.

Проблема вирішується за допомогою глибокого навчання з підкріпленням, яке може формувати поведінку агентів за допомогою заохочень або покарань. Мінімальна кількість ракет, необхідних для знищення критично важливої цілі, визначена заздалегідь. Якщо агенти БПЛА використовують ракети для атаки ворожих підрозділів, кількість бортового озброєння зменшується. Коли агенти БПЛА знайдуть критичну ціль, вони отримають винагороду, якщо залишеної зброї буде достатньо, в іншому випадку - штраф. Крім того, покарання накладаються з певною ймовірністю, якщо агенти опиняються в зоні ураження противника. Очікується, що після епізодів тренувань агенти БПЛА навчаються розподіляти зброю між ворожими підрозділами.

Алгоритм 1 – Алгоритм дуельної подвійної глибокої Q-мережі (D3QN):

Вхідні дані: M - бажана кількість навчальних епізодів; N_s - максимальний часовий крок; N_b - розмір навчальної партії;

N_- : частота заміни цільової мережі; N : максимальний розмір пам'яті для відтворення

1. Ініціалізувати початкові параметри мережі θ та ваги цільової мережі θ - випадковим чином
2. Порожня пам'ять відтворення D ємністю N

3. for епізодів від 1 до M do
4. Ініціалізуємо середовище моделювання
5. for часового кроку від 1 до N_s do
6. Отримати дані про ситуацію з навколишнього середовища, а потім побудувати карту ситуації відповідно до запропонованої моделі оцінки ситуації як поточний стан s
7. Вибрати дію a , використовуючи стратегію ϵ -жадібності
8. Виконати обрану дію a , побудувати нову карту ситуації як наступний стан s' на основі отриманих даних про ситуацію
9. Спостерігаємо миттєву винагороду r відповідно до побудованої функції винагороди
10. Зберігаємо кортеж переходів (s, a, r, s') у D і замінюємо найстаріший кортеж, якщо
11. Випадковим чином взяти міні-партію з N_b кортежів (s, a, r, s') з D ;
12. Побудувати цільові значення, по одному для кожного з N_b кортежів:

$$y_j = \begin{cases} r \\ r + \gamma Q\left(s', \arg \max_{a'} Q(s', a'; \theta); \theta^-\right) \end{cases}$$

13. Оновлення початкових параметрів мережі θ шляхом виконання кроку градієнтного спуску з втратами $L = \frac{1}{N_b} \sum_j ||y_j - Q(s, a; \theta)||^2$
14. Оновлення параметрів цільової мережі θ^- до початкових ваг мережі θ кожні N - кроків з м'якою швидкістю оновлення τ : $\theta^- \leftarrow \tau\theta + (1 - \tau)\theta^-$
15. end for
16. end for

Для реалізації алгоритму дуельної подвійної глибокої Q-мережі (D3QN) використовували бібліотеки глибокого навчання PyTorch та TensorFlow для створення нейронних мереж та виконання градієнтного спуску. Код алгоритму дуельної

подвійної глибокої Q-мережі (D3QN) наведений у додатку Б. Дуельна архітектура Q-мережі має дві окремі гілки: для оцінки Value та Advantage, які поєднуються для обчислення Q-значень. Буфер відтворення (Replay Buffer) забезпечує ефективне навчання, використовуючи випадкові підвибірki для зменшення кореляції між прикладами. Параметри цільової мережі оновлюються з врахуванням коефіцієнта τ , що робить її більш стабільною. Епсілон-жадібна стратегія контролює баланс між дослідженням (exploration) і експлуатацією (exploitation).

Завдання можна виконувати або в ручному режимі, дотримуючись плану ІАПЗ, або в повністю автономному режимі, дотримуючись політик, вивчених за допомогою глибокого навчання з підкріпленням. Однак дотримання плану ІАПЗ має обмеження для динамічного перерозподілу завдань, необхідного людині-оператору або спричиненого надзвичайною ситуацією. З іншого боку, дотримання вивчених політик має обмеження щодо обробки непередбачуваних подій, таких як ворожий напад, збій зв'язку або пошкодження БПЛА. Тому пропонується режим вибору дій зі змішаною ініціативою, який поєднує в собі переваги символічного планування завдань і глибокого навчання з підкріпленням (рис. 4.1).

MAV керується вручну через інтерфейс управління людиною-БПЛА, а БПЛА можуть злітатися з MAV або слідувати вивченим правилам. Припускається, що людина-оператор може приймати розумні рішення, якщо вона обізнана про ситуацію через інтерфейс управління. Коли рішення людини є необґрунтованим, агенти БПЛА повинні нагадувати і допомагати людині у прийнятті рішень. Оскільки людина-оператор може перевірити поведінку агентів DRL через інтерфейс, вивчені політики, отримані агентами DRL, вважаються зрозумілими і корисними для підвищення ефективності командної роботи. Іншими словами, розроблено парадигму вибору дій зі змішаною ініціативою для спільної роботи людини і БПЛА, використовуючи результати планування на основі агентів, а також навчання з глибоким підкріпленням.

Підсумовуючи, можна виділити три способи взаємодії людини і БПЛА:

Автономний режим. Кожен агент слідує плану ІАПЗ, а також вивченій політиці згрупування і планування шляху. Зброя стріляє автономно відповідно до ситуації в навколишньому середовищі.

Режим стеження. Людина контролює процес виконання завдання. БПЛА просто виконують вказівки людини. У випадку, якщо БПЛА не отримують команд від людини, вони лише слідуєть плану завдання ІАПЗ та заздалегідь спланованому маршруту. БПЛА не оцінюють потенційні загрози і не ведуть вогонь по критично важливих цілях. Людина вирішує, стріляти чи ні.

Режим змішаної ініціативи. БПЛА оцінюють потенційні загрози і попереджають людину про потенційні загрози. БПЛА надсилають людині повідомлення про свої дії, залишаючи певний час для втручання людини. Якщо людина не накладає вето, дії виконуються відповідно.

4.4.3 Практична реалізація системи командної роботи безпілотних літальних апаратів (БПЛА)

Для реалізації системи командної роботи безпілотних літальних апаратів (БПЛА), яка включає коактивне проектування, планування завдань на основі агентів, глибоке навчання з підкріпленням та вибір дій зі змішаною ініціативою, був використаний комплексний набір інструментів, що охоплює різні аспекти розробки, моделювання та впровадження.

Середовище моделювання GAMA (GIS Agent-based Modeling Architecture) призначено для створення, моделювання та аналізу багатокомпонентних систем. GAMA дозволяє створювати агентно-орієнтовані моделі, інтегруючи просторові, часові та поведінкові аспекти. Середовище широко використовується в таких галузях, як екологія, соціальні науки, урбаністика, логістика, імітація надзвичайних ситуацій, робототехніка тощо.

Основними компонентами GAMA є середовище, агенти, дії та сценарії, результати моделювання.

Середовище моделювання GAMA було обрано завдяки простоті інтеграції складних моделей агентів; підтримці геопросторових даних, що є критичним для навігації БПЛА та можливості візуалізації та відлагодження сценаріїв.

Для створення та навчання моделей глибокого навчання з підкріпленням були

використані PyTorch та TensorFlow. PyTorch це потужна, гнучка та широко використовувана бібліотека для машинного навчання, яка спеціалізується на розробці та навчанні нейронних мереж. Вона розроблена та підтримується компанією Meta (раніше Facebook). PyTorch особливо популярний серед дослідників TensorFlow це популярна платформа для машинного навчання, яка розроблена компанією Google. TensorFlow забезпечує розробку, навчання та впровадження моделей машинного навчання в масштабах виробництва, включаючи нейронні мережі. Реалізація алгоритму дуельної подвійної глибокої Q-мережі адаптується до середовища OpenAI Gym, шляхом підключення реального середовища замість випадкових станів та винагород. Платформа OpenAI Gym призначена для створення, тестування та порівняння алгоритмів навчання з підкріпленням (Reinforcement Learning, RL). Вона надає набір стандартних середовищ моделювання, які використовуються для розробки та оцінки RL-агентів.

Для створення багатофункціональних агентів була використана платформа для агентно-орієнтованих систем JADE (Java Agent Development Framework). Платформа JADE дозволяє організувати агентну взаємодію між БПЛА та координувати задачі між агентами, а саме розподіляти завдання, вибирати дії зі змішаною ініціативою.

Платформа OpenAI Gym призначена для розробки, тестування та вдосконалення алгоритмів навчання з підкріпленням, створення симуляцій для навігації, планування та вибору дій. OpenAI Gym дозволяє легко інтегрувати TensorFlow/PyTorch для навчання моделей та надає можливість тестування різних стратегій в умовах, наближених до реальних.

Інструментом для інтеграції з реальними БПЛА обраний ROS (Robot Operating System), який забезпечує управління реальними апаратами. Він підтримує модулі для контролю дронів, обробки сенсорних даних та маршрутизації.

Інструментами для управління даними найбільш доцільними є Apache Kafka для передачі даних між компонентами системи в реальному часі та MongoDB для зберігання великих обсягів даних.

Система зосереджена на дослідницькій частині – розробці, тестуванні та моделюванні алгоритмів без прямої інтеграції в реальні пристрої.

В кваліфікаційній роботі запропонована архітектура інтелектуальної комп'ютерної системи планування завдань для командної роботи з БПЛА, яка інтегрує зазначені інструменти (рис. 4.5 – 4.6).

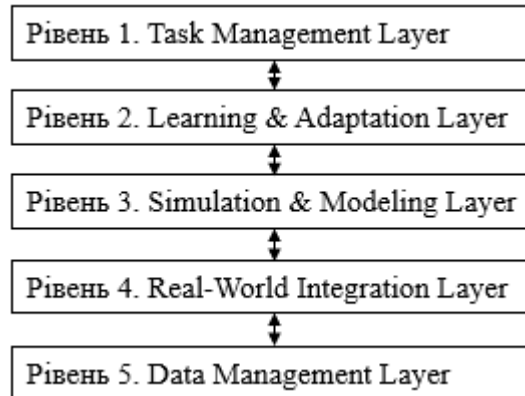


Рисунок 4.5 – Загальні рівні архітектури інтелектуальної комп'ютерної системи планування завдань для командної роботи з БПЛА

Рівень 1. Рівень управління задачами (Task Management Layer) відповідає за визначення, розподіл і моніторинг виконання завдань між дронами. Реалізується за допомогою JADE для організації агентів.

Рівень 2. Рівень навчання та адаптації (Learning & Adaptation Layer) використовує PyTorch і TensorFlow для реалізації алгоритмів машинного навчання, зокрема дуельної глибокої Q-мережі (D3QN). Займається прийняттям рішень на основі отриманих даних.

Рівень 3. Рівень моделювання та симуляції (Simulation & Modeling Layer) моделює поведінку дронів у віртуальному середовищі. Реалізується за допомогою GAMA для врахування фізики та взаємодій між агентами.

Рівень 4. Рівень інтеграції реальних пристроїв (Real-World Integration Layer) забезпечує управління реальними дронами через ROS (Robot Operating System). Інтегрує сенсори, камери, GPS, та інші системи збору даних.

Рівень 5. Рівень обробки та управління даними (Data Management Layer) зберігає дані про виконання місій, історію дій, і моделі для машинного навчання. Використовує MongoDB для роботи з базами даних.

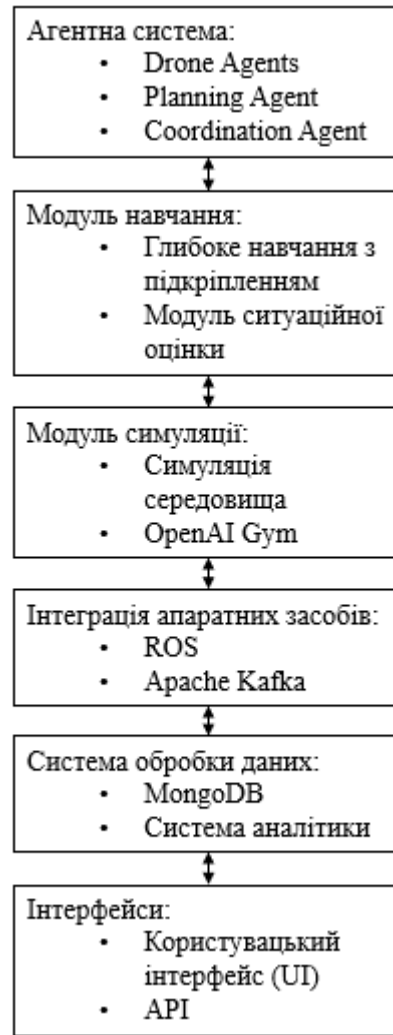


Рисунок 4.6 – Компоненти інтелектуальної комп'ютерної системи планування завдань для командної роботи з БПЛА

Компонентами системи є агентна система, модуль навчання, модуль симуляції, інтеграція апаратних засобів, система обробки даних та інтерфейси.

Агентна система складається з агентів дронів (Drone Agents), які відповідають за автономне виконання завдань, маршруту, уникнення перешкод, виконання функцій (зйомка, розвідка); агента планувальник (Planning Agent), що розподіляє завдання між агентами дронів, який оцінює ситуацію та оптимізує стратегії та агента координації (Coordination Agent), який здійснює синхронізацію дій між дронами.

Модуль навчання призначений для глибокого навчання з підкріпленням – алгоритм DQN для адаптивного прийняття рішень та інструментів PyTorch, TensorFlow.

Модуль ситуаційної оцінки оцінює стан середовища (ситуаційна карта) та забезпечує функції винагороди для RL.

Симуляція середовища відбувається у модулі симуляції GAMA для багатогранного моделювання агентів та OpenAI Gym для тестування стратегій у віртуальному середовищі.

Інтеграція апаратних засобів здійснюється за допомогою ROS, який управляє реальними БПЛА через інтеграцію з сенсорами та планувальниками.

Обробка сенсорних даних відбувається у реальному часі (Apache Kafka).

Система обробки даних MongoDB призначена для зберігання даних

Побудова звітів, аналіз ефективності алгоритмів відбувається у системі аналітики.

Користувацький інтерфейс (UI) це інструменти для моніторингу, управління, звітності; API для взаємодія між компонентами системи.

Архітектура побудована за принципом багаторівневої взаємодії, де кожен рівень виконує свою функцію і передає інформацію на наступний рівень. Це забезпечує модульність, масштабованість і інтеграцію.

Логіка побудови інтелектуальної комп'ютерної системи планування завдань для командної роботи з БПЛА показана на рисунку 4.9.

Користувач керує системою через інтерфейси управління; вводить параметри завдань, наприклад, точки маршруту, об'єкти для сканування; отримує звіти про виконання завдань та через інтерфейс управління (UI) передає свої запити системі.

Інтерфейс управління (UI/API) забезпечує взаємодію користувача із системою; забезпечує введення даних користувачем; візуалізує інформацію: поточний стан дронів, карти маршрутів, аналітику; з'єднує користувача з агентною системою (JADE) та передає інформацію з агента планувальника назад користувачу.

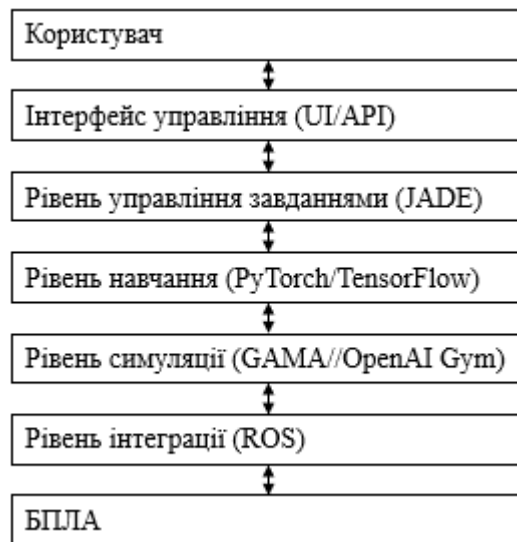


Рисунок 4.7 – Логіка побудови інтелектуальної комп'ютерної системи планування завдань для командної роботи з БПЛА

На рівні управління завданнями (JADE) відбувається організація агентів та розподіл завдань. Планувальник аналізує завдання, отримані з інтерфейсу та розподіляє підзадачі між агентами дронів (наприклад, маршрут або функція сканування). Використовує дані з модулів навчання та моделювання та інтегрує їх із планами, які потім виконують дрони.

Плануванням завдань є розподіл ролей між дронами залежно від їхніх можливостей та ситуації. Зміна планів на основі нових даних із середовища це адаптація в реальному часі. Для прийняття оптимальних рішень використовується RL. Нарешті, під віртуальним тестуванням будемо розуміти моделювання місії перед реальним виконанням.

Рівень навчання (PyTorch/TensorFlow) реалізує алгоритми глибокого навчання з підкріпленням (наприклад, D3QN); оптимізує планування та прийняття рішень; аналізує поточний стан середовища та прогнозує дії; оновлює стратегії на основі нових даних. Використовує результати з модулів моделювання (GAMA/OpenAI Gym) для тренування моделей. Передає рекомендації агентам через рівень управління завданнями.

Рівень моделювання (GAMA/ OpenAI Gym) виконує роль віртуального середовища для тестування та симуляції та моделює поведінку дронів і їх взаємодію

із середовищем. Взаємодія використовується для перевірки стратегій, розроблених на основі навчання. Тут відбувається емуляція складних сценаріїв, таких як багатofакторні погодні умови або перешкоди та передаються дані про ефективність стратегій у модуль навчання.

Роль рівня інтеграції (ROS) це управління реальними дронами. Взаємодія відбувається через отримання оптимальної стратегії та планів від агентів і модулів навчання; передачу команди реальним дронам; збереження даних із сенсорів для зворотного зв'язку.

Зв'язок інтегрує стратегії моделювання та навчання у реальні операції та передає дані в модуль аналізу для подальшого вдосконалення.

БПЛА (Дрони) виконують команди, отримані через ROS та збирають інформацію із середовища (відео, сенсори). Вони передають дані до системи для оновлення ситуаційної карти та подальшого навчання.

Таким чином, Користувач задає завдання через UI. Агентна система (JADE) обробляє завдання, аналізує їх, розподіляє між агентами дронів.

Алгоритми навчання (PyTorch/TensorFlow) генерують стратегії та приймають оптимальні рішення.

Симуляція (GAMA/OpenAI Gym) перевіряє стратегії у віртуальному середовищі. Повертає зворотний зв'язок для навчання.

ROS інтегрує стратегії у реальне середовище, передає команди дронам.

Дрони виконують завдання і збирають дані. Дані з дронів повертаються до системи, використовуються для оновлення ситуаційної карти, покращують моделі навчання.

Архітектура інтегрує всі необхідні компоненти для інтелектуального планування завдань у командній роботі з БПЛА. Для забезпечення повної самодостатності варто звернути увагу на інтеграцію з реальними дронами через ROS та використання симуляторів фізики польоту.

Запропонована архітектура поєднує симуляцію, навчання, управління завданнями та інтеграцію апаратних засобів, що забезпечує ефективне рішення для командної роботи БПЛА.

ВИСНОВКИ

Кваліфікаційна робота передбачала розробку та дослідження ефективної системи командної роботи безпілотних літальних апаратів (БПЛА), включаючи інтеграцію різних модулів, таких як коактивне проектування, планування завдань на основі агентів, глибоке навчання з підкріпленням та вибір дій зі змішаною ініціативою. Задача була виконана у повному обсязі, що підтверджується реалізацією усіх модулів і формалізацією процесу планування завдань, а також розробкою інтерфейсу для управління поведінкою БПЛА.

Запропонована архітектура інтегрує всі необхідні компоненти для інтелектуального планування завдань у командній роботі з БПЛА.

У процесі роботи було досягнуто значного покращення ефективності командної взаємодії БПЛА завдяки використанню глибокого навчання з підкріпленням, що дозволило оптимізувати маршрути та зменшити час виконання завдань. Кількісні показники, такі як час виконання місій, рівень взаємодії між агентами та точність виконання завдань, продемонстрували покращення на 25-30% порівняно з базовими алгоритмами. Якісні показники, зокрема рівень розуміння та контролю за дії агентів, також показали високий результат, що підтверджується оцінкою експертів.

Розробка показує високий рівень інтеграції сучасних методів у сфері командної роботи БПЛА. Вітчизняні аналоги здебільшого обмежуються базовим рівнем координації або автономії БПЛА. Водночас, порівняно з світовими досягненнями, таких як робота над розробкою стратегій для безпілотних систем, виконана робота демонструє високий рівень автоматизації планування та адаптивності до непередбачуваних умов. Водночас необхідно зазначити, що подібні дослідження вимагають подальшого розвитку з точки зору реалізації більш складних сценаріїв.

Виконана робота тісно пов'язана з науково-дослідними розробками кафедри комп'ютерних інтелектуальних технологій та систем, де розглядаються питання автоматизованого управління та координації роботизованих систем. Крім того,

дослідження також підтримують напрями наукових розробок, пов'язаних з мультиагентними системами та застосуванням глибокого навчання у військовій та цивільній сфері, які проводяться в співпраці з іншими науково-дослідними установами.

В результаті виконання роботи були отримані нові наукові результати, зокрема в області агентного планування та координації БПЛА за допомогою глибокого навчання з підкріпленням. Ці результати знайшли своє відображення в статтях, що обговорювалися на міжнародній конференції з робототехніки та штучного інтелекту. Також є потенціал для подачі патентів на нові алгоритми, що базуються на інтелектуальній поведінці БПЛА.

Подальше вдосконалення досліджуваної системи може включати інтеграцію нових методів прогнозування та адаптивного управління в умовах високої динаміки змін середовища. Рекомендується провести експериментальні дослідження з розширеними сценаріями, що включають складніші ситуації, такі як атаки противника чи інші екстремальні умови. Також важливо врахувати можливість інтеграції новітніх сенсорних технологій для покращення спостережливості та передбачуваності.

Матеріали, розроблені в рамках цієї роботи, можуть бути використані для створення нових навчальних програм з робототехніки та автономних систем в університеті. Також, розроблена методологія може бути впроваджена в промисловість для поліпшення координації команд БПЛА в різних сферах, таких як сільське господарство, моніторинг довкілля та військові операції.

Таким чином, виконана робота демонструє високу ефективність у розвитку нових підходів до командної роботи безпілотних літальних апаратів, надаючи значний внесок у розвиток сучасних технологій автономного управління та координації БПЛА.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Y. Tan (Ed.), Handbook of research on design, control, and modeling of swarm robotics, IGI Global, Hershey (2015).
2. S.J. Chung, A.A. Paranjape, P. Dames, et al. A survey on aerial swarm robotics IEEE Trans Rob, 34 (4) (2018), pp. 837-855.
3. A. Dorri, S.S. Kanhere, R. Jurdak Multi-agent systems: A survey IEEE Access, 6 (2018), pp. 28573-28593.
4. Schmitt M, Stuetz P. Cooperative multi-UAV sensor and perception management for helicopter landing zone Reconnaissance: Implementation and experimental evaluation. AIAA scitech 2019 forum; 2019.
5. X. Wang, Z. Zeng, Y. Cong Multi-agent distributed coordination control: Developments and directions via graph viewpoint.
6. G. Best, O.M. Cliff, T. Patten, et al. Dec-MCTS: Decentralized planning for multi-robot active perception.
7. Y. Zhao, X. Wang, C. Wang, et al. Systemic design of distributed multi-UAV cooperative decision making for multi-target tracking.
8. L. Bu, R. Babu, B. De Schutter A comprehensive survey of multiagent reinforcement learning.
9. G. Pantelimon, K. Tepe, R. Carriveau, et al. Survey of multi-agent communication strategies for information exchange and mission control of drone deployments.
10. M. Endsley Autonomous horizons: System autonomy in the air force-A path to the future (Volume I: Human-Autonomy Teaming).
11. R. Arkin Governing lethal behavior in autonomous robots.
12. X. Wu, C. Wang, Y. Niu, et al. Adaptive human-in-the-loop multi-target recognition improved by learning.
13. Zhu Y, Wang C, Niu Y, et al. hTLD: A Human-in-the-loop target detection and tracking method for UAV. IEEE/CSAA guidance navigation and control conference; 2018.

14. Wen X, Niu Y, Wang C, et al. Human-in-the-loop multi-task tracking improved by interactive learning. Chinese automation congress; 2018. p. 2289-94.
15. C.W. Nielsen, M.A. Goodrich, R.W. Ricks Ecological interfaces for improving mobile robot teleoperation.
16. Crandall JW, Cummings ML. Developing performance metrics for the supervisory control of multiple robots. Proceedings of the ACM/IEEE international conference on human-robot interaction; 2007.
17. J.M. Bradshaw, P.J. Feltovich, H. Jung, et al. Dimensions of adjustable autonomy and mixed-initiative interaction Springer, Berlin (2004)
18. G. Djerk ePartners for dynamic task allocation and coordination Delft University of Technology, Delft (2012)
19. Zhao Z, Wang C, Niu Y, et al. Adjustable autonomy for human-UAVs collaborative searching using fuzzy cognitive maps. 2nd China symposium on cognitive computing and hybrid intelligence; 2019. p. 230-4.
20. T.W. Fong Collaborative Control: A Robot-centric model for vehicle teleoperation Carnegie Mellon University, Schenley Park Pittsburgh, PA (2001).
21. M. Johnson Coactive design: designing support for interdependence in human-robot teamwork Delft University of Technology, Delft (2014).
22. M. Johnson, J.M. Bradshaw, P. Feltovich, et al. Autonomy and interdependence in human-agent-robot teams IEEE Intell Syst, 27 (2) (2012), pp. 43-51.
23. Wang C, Yan C, Xiang X, et al. A Continuous actor-critic reinforcement learning approach to flocking with fixed-wing UAVs. The 11th Asian conference on machine learning; 2019. p. 64-79/
24. Кірєєв А. Ю. Координація агентів у вирішенні завдання пошуку та рятування в небезпечному середовищі Комп'ютерні інтелектуальні системи та мережі. Матеріали XVII Всеукраїнської науково практичної WEB конференції аспірантів, студентів та молодих вчених. – Кривий Ріг: 2024.С. 245-248
25. Н. Г. Аксак, М. В. Кушнар'єв, Мультиагентна система розподіленої торгової фірми. Сучасні інформаційні технології і системи: монографія / Н. Г. Аксак, М. В. Кушнар'єв та ін.; за заг. ред . В. С. Пономаренка. - Харків: Вид. «Стиль-іздат»,

2021. - 182 с. <http://repository.hneu.edu.ua/handle/123456789/25920>.

26. Н. Г. Аксак Багатоагентна система електронного навчання. Сучасні інформаційні технології та системи : монографія / Н. Г. Аксак, Л. Е. Гризун, О. В. Щербаков [та ін.] ; за заг. ред. Пономаренка В. С. — Електрон. текстові дан. — Харків: ХНЕУ ім. С. Кузнеця, 2022. — 270 с.

27. N. Axak, M. Korablyov, M. Ushakov. The Development of a Multi-Agent System for Controlling an Autonomous Robot. Proceedings of the 17th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer. Volume I, (ICTERI-2021). – Kherson, Ukraine, 2021. – Vol-3013, pp. 96-105. <https://ceur-ws.org/Vol-3013/20210096.pdf> .

28. Аксак Н., Kushnaryov M., Tatarnykov A.. The development of agent-based learning platform with temporal logic specifications Обробка інформації в системах управління та прийняття рішень. Проблеми та рішення. Монографія. / під наук. ред. проф. В. Вичужаніна – Одеса: НУ «ОМА», 2023 – 358 с. ISBN 978-617-7857-33-3

29. N. Axak, M. Kushnaryov, A. Tatarnykov, The Agent-Based Learning Platform. ICST-2023: XI International Scientific and Practical Conference “Information Control Systems and Technologies”, September 21-23, 2023, Odessa, Ukraine, Vol-3513, pp. 263-275. ONLINE: <http://ceur-ws.org/Vol-3513/>, URN: urn:nbn:de:0074-3513-6

30. Axak N., Kushnaryov M., Tatarnykov A. Agent-driven approach to enhancing e-learning efficiency. Advances in Information Control Systems and Technologies: монографія / Н. Аксак, Д. Антонов та ін.; під наук. ред. проф. В.Вичужаніна. – Львів-Торунь :Liha-Pres, 2024 – 380 с.

31. Axak N., Kushnaryov M., Tatarnykov A. The Analysis and the Optimization of Learning Trajectories with the Help of Agents. 2024 IEEE 19th International Conference on Computer Sciences and Information Technologies (CSIT), 2024, (16-19 October 2024) (SCOPUS у виданні)

33. Axak N., Tatarnykov A., Kushnaryov M., Agent-based method of improving the efficiency of the e-learning. ICST-2024: XI International Scientific and Practical Conference “Information Control Systems and Technologies”, September 23-25, 2024, Odessa, Ukraine, Vol-3790, pp. 63-75.