Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)
Кафедра Штучного інтелекту
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Дослідження та застосування методів ройового інтелекту
у задачі керування дронами-зграєю
(тема)

Виконав:
здобувач другого року навчання,
групи СШІм-24-2
Віктор ЛАЗАРЄВ
(власне ім'я, прізвище)
Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)
Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)
Освітня програма Системи штучного інтелекту
(повна назва освітньої програми)
Керівник проф. Наталія РЯБОВА
(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри ШІ _____ Лариса ЧАЛА
(підпис) (власне ім'я, прізвище)

2026 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук

Кафедра Штучного інтелекту

Рівень вищої освіти другий (магістерський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

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

Освітня програма Системи штучного інтелекту
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«_____» _____ 20 ___ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Лазарєву Віктору Олексійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження та застосування методів ройового інтелекту у задачі керування дронами-зграєю

затверджена наказом університету від 6 квітня 2026 р. № 269Ст
2. Термін подання здобувачем роботи до екзаменаційної комісії 20 травня 2026 р.
3. Вихідні дані до роботи Науково-технічні публікації, дані Інтернет-джерел та відомих наукових проєктів, документація мови програмування Java, документація фреймворку JADE, документація мови програмування Python

4. Перелік питань, що потрібно опрацювати в роботі
1) Огляд стану проблеми та постановка завдань дослідження
2) Обґрунтування вибору технологічних рішень
3) Методи та алгоритми розв'язання завдання керування зграєю
4) Програмна реалізація спеціалізованої багатоагентної системи

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / термін виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	06.04.2026	виконано
2	Дослідження актуальності теми	09.04.2026	виконано
3	Аналіз предметної області	13.04.2026	виконано
4	Огляд існуючих рішень	15.04.2026	виконано
5	Написання пояснювальної записки	08.05.2026	виконано
6	Перевірка на академічний плагіат	14.05.2026	виконано
7	Нормоконтроль	15.05.2026	виконано
8	Рецензування	15.05.2026	виконано
9	Підготовка презентації та доповіді	19.05.2026	виконано
10	Захист перед ЕК	20.05.2026	

Дата видачі завдання 6 квітня 20 26 р.

Здобувач _____
(підпис)

Керівник роботи _____ проф. Наталія РЯБОВА
(підпис) (посада, власне ім’я, прізвище)

РЕФЕРАТ

Пояснювальна записка: 73 с., 22 рисунок, 2 табл., 1 дод., 39 джерел.

АГЕНТНИЙ ШІ, АРХІТЕКТУРА, БПЛА, ВИБУХОНЕБЕЗПЕЧНІ ОБ'ЄКТИ, ЗГРАЯ ДРОНІВ, ІНТЕЛЕКТУАЛЬНИЙ АГЕНТ, КОГНІТИВНА МОДЕЛЬ, МУЛЬТИАГЕНТНА СИСТЕМА, НЕЙРОННА МЕРЕЖА, РОЙОВИЙ ІНТЕЛЕКТ.

Об'єкт дослідження – мультиагентна системи для керування дронами-зграєю на основі інтелектуальних агентів.

Предмет дослідження – методи ройового інтелекту та архітектури когнітивних моделей для керування дронами-зграєю.

Мета роботи – дослідження та застосування методів ройового інтелекту для керування дронами-зграєю в реальному часі.

Методи дослідження – теоретичні: методи ройового інтелекту, комп'ютерного зору, машинного та глибокого навчання, обробки й аналізу даних; практичні: класифікація, нейронні мережі, концептуальна модель, архітектури агентів та мультиагентної системи керування, розробка програмного коду.

Результатом роботи є програмна реалізація моделі керування дронами-зграєю на основі створення гібридного алгоритму з ієрархічною розподіленою архітектурою системи, з використанням агентного штучного інтелекту та ройових методів оптимізації. Забезпечується висування зграї на визначені позиції, пошук та виявлення об'єктів, передача даних, їх зберігання та візуалізація, а також підтримання стабільного зв'язку.

Перспективними напрямками впровадження розробленої мультиагентної системи є гуманітарні місії розмінування, рятувальних операцій, що є нагальною необхідністю сьогодення, а також пошукових та дослідницьких робіт.

ABSTRACT

Master’s thesis contains: 73 pp., 22 fig., 2 tabl., 1 ann., 39 references.

AGENTIC AI, ARCHITECTURE, COGNITIVE MODEL, DRONE SWARM, EXPLOSIVE OBJECTS, INTELLECTUAL AGENT, MULTI-AGENT SYSTEM, NEURAL NETWORK, SWARM INTELLIGENCE, UAV.

The object of research is a multi-agent system for controlling drone flocks based on intelligent agents.

The subject of research is swarm intelligence methods and the architecture of cognitive models for controlling drone flocks.

The purpose of research is to study and apply swarm intelligence methods for controlling drone swarms in real time.

Research methods are theoretical: methods of swarm intelligence, computer vision, machine and deep learning, data processing and analysis; practical: classification, neural networks, conceptual model, agent architectures and multi-agent control system, development of software code.

The result of the work is a software implementation of a drone swarm control model based on the creation of a hybrid algorithm with a hierarchical distributed system architecture, using agent-based artificial intelligence and swarm optimization methods. The swarm is deployed to specified positions, search and detection of objects, data transfer, storage and visualization, as well as maintaining stable connection.

Promising areas for the implementation of the developed multi-agent system are humanitarian missions of demining, rescue operations, which are an urgent need of today, as well as search and research work.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів 8
Вступ 9
1 Огляд стану проблеми та постановка завдань дослідження 12
1.1 Огляд методів керування БПЛА 12
1.2 Мультиагентний метод інтелектуалізації керування 13
1.2.1 Основні компоненти когнітивної архітектури 14
1.2.2 Методи інтелектуалізації агентів 17
1.2.3 Сучасні напрямки розвитку інтелектуальних агентів 19
1.3 Задача керування зграєю як когнітивна модель 20
1.4 Мета та постановка завдань дослідження 23
2 Обґрунтування вибору технологічних рішень 24
2.1 Методи та моделі ройового інтелекту 24
2.1.1 Particle Swarm Optimization 24
2.1.2 Ant Colony Optimization 26
2.1.3 Artificial Bee Colony 27
2.1.4 Модель Boids 27
2.2 Порівняльний аналіз типів архітектур агентів 28
2.2.1 Прості рефлекторні агенти (Simple Reflex Agents) 28
2.2.2 Рефлекторні агенти на основі моделей (Model-based Reflex Agents) 30
2.2.3 Агенти на основі цілей (Goal-based Agents) 31
2.2.4 Агенти на основі корисності (Utility-based Agents) 33
2.2.5 Навчальні агенти (Learning Agents) 33
2.2.6 Порівняльний аналіз архітектурних підходів 35
2.3 Аналіз сценарію вирішення задачі керування зграєю 37

2.4 Вибір технологічних рішень 39
3 Методи та алгоритми розв’язання задачі керування зграєю 40
3.1 Архітектура агентів 40
3.1.1 Агент-пошуковець 40
3.1.2 Агент-зв’язківець 44
3.1.3 Агент-координатор 46
3.1.4 Навігація агентів 48
3.2 Архітектура та алгоритм роботи мультиагентної системи 49
4 Програмна реалізація спеціалізованої багатоагентної системи 53
4.1 Технології та інструменти 53
4.2 Архітектура програмного забезпечення 54
4.3 Програмна реалізація системи керування 55
4.3.1 Модуль навігації та уникнення зіткнень 57
4.3.2 Модуль зв’язку та ретрансляції 58
4.3.3 Протокол обміну даними 59
4.4 Інтерфейс користувача 60
4.5 Моделювання керування зграєю дронів при пошуку об’єктів 61
4.6 Аналіз результатів моделювання 64
Висновки 65
Перелік джерел посилання 68
Додаток А Відомість кваліфікаційної роботи 73

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

БПЛА – безпілотний літальний апарат;

ПЗ – програмне забезпечення;

ПК – персональний комп’ютер;

ШІ – штучний інтелект;

ABC – Artificial Bee Colony – штучний алгоритм бджолиної сім’ї;

ACO – Ant Colony Optimization – алгоритм оптимізації мурашиної колонії;

ACL – Agent Communication Language – мова комунікації агентів;

AI – Artificial Intelligence – штучний інтелект;

API – Application Programming Interface – інтерфейс прикладного програмування;

CNN (ConvNet) – Convolutional Neural Network – конволюційна нейронна мережа;

FIPA – Foundation for Intelligent Physical Agents – стандарт комп’ютерного програмного забезпечення для взаємодіючих агентів;

GPS – Global Positioning System – супутникова система навігації;

JADE – Java Agent Development Framework – програмний каркас розробки мультиагентних систем;

LLM – Large Language Model – велика лінгвістична модель;

PSO – Particle Swarm Optimization – метод рою часток;

UDP – User Datagram Protocol – протокол швидкої передачи даних без встановлення з’єднання;

YOLO – You Only Look Once – модель нейромереж для швидкого виявлення об’єктів на зображеннях;

2D – двовимірний простір;

3D – тривимірний простір.

ВСТУП

Актуальність теми. Галузь безпілотних авіаційних систем та проблема керування такими системами є надзвичайно актуальною. Впродовж останніх років цей напрямок розвивається швидкими темпами як в Україні, так і в світі для вирішення різного типу задач, серед яких ряд гуманітарних застосувань [1]:

– пошукові, рятувальні операції під час надзвичайних подій, стихійних лих та в важкодоступній місцевості;

– виявлення вибухонебезпечних предметів, контроль якості розмінування територій та безпеки користування;

– протипожежний та протидиверсійний моніторинг великих масивів лісових угідь;

– моніторинг та внесення засобів захисту рослин на полях, тощо.

Є різні підходи до вирішення цих проблем. При традиційному методі використання ручного труда на місцевості для обстеження великих ділянок потребує великої кількості людського ресурсу. До того ж існує ризик, зокрема при виявленні небезпечних предметів, при ліквідації наслідків стихійних лих, контролю зон обмеженого доступу, тощо. Серед дистанційних методів – розпізнавання оператором на основі візуального спостереження з повітряних транспортних засобів (гелікоптери, легкомоторні літаки), що значно підвищує вартість та має низьку точність виявлення. При застосуванні безпілотних засобів традиційно один оператор керує лише одним засобом, адже типові системи передбачають ручне керування БПЛА. При масштабних операціях виникає потреба у великій кількості операторів, що призводить до збільшення витрат та вимагає координації дій.

Це демонструє потребу в інтелектуалізації процесів пошуку із застосуванням новітніх методів та технологій. Для роботи у складних умовах природного середовища існує ряд сучасних підходів на основі

використання агентного метода, що знайшли широке використання як у цивільних, так і військових цілях завдяки значним перевагам у порівнянні з традиційними методами:

– безпека персоналу;

– низька вартість;

– відносна автономність та мобільність;

– швидкість обробки великих територій;

– швидке прийняття рішень щодо виявлених результатів;

– можливість інтеграції з додатковим обладнанням.

Серед існуючих технічних рішень розробка мультиагентних систем є одним з перспективних. Цей напрямок дослідження привертає увагу завдяки високій ефективності, економії часу, гнучкості системи та легкості масштабування. Застосування систем на основі штучного інтелекту з відносною автономністю є актуальним для колективного розв’язання різноманітних завдань. Така технологія вже застосовується для керування в розподілених динамічних системах з декількома інтелектуальними агентами та паралельними обчислювальними процесами. При цьому агенти як самостійні спеціалізовані комп’ютерні програми здатні викликати потрібні інструменти в процесі роботи, передавати й зберігати завдання та дані.

Кваліфікаційна робота є продовженням досліджень проблеми гуманітарного розмінування, розпочатих в бакалаврській роботі. З огляду на розвиток технологій розглядається створення інтелектуальної багатоагентної системи для заміни небезпечного й затратного процесу засобами на основі штучного інтелекту для пошуку й фіксації небезпечних предметів. Розроблена технологія здатна забезпечити обробку великих площ, опрацювання отриманих даних з плануванням кроків, виявлення небезпечних об’єктів чи здійснення контролю якості розмінованих територій.

В роботі описана побудова системи керування дронами-зграєю із застосуванням методів ройового інтелекту для руху, інтелектуалізації процесу виявлення на основі агентного штучного інтелекту, взаємодії агентів, що сумісними зусиллями сприяють ефективному виконанню завдання. Результати роботи можуть застосовуватись в системах гуманітарного розмінування, а також для виконання пошукових та рятувальних операцій, що є нагальною необхідністю сьогодення.

1 ОГЛЯД СТАНУ ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАВДАНЬ ДОСЛІДЖЕННЯ

1.1 Огляд методів керування БПЛА

За класичною архітектурою способи керування БПЛА обмежені використанням пульту або автопілоту з визначеними функціями [2], чого недостатньо для побудови складних систем. Ручне керування має ряд недоліків [3]:

– потреба в операторах;

– висока вартість;

– низька ефективність через «людський фактор»;

– нестабільність зв’язку, що є одним з обмежуючих факторів.

До того ж деякі завдання за відсутності автономного алгоритму втрачають свою ефективність. Одним з варіантів вирішення цієї проблеми є комплекс керування БПЛА «Віщун-К1» [4], що вміщує наземну станцію, засоби зв’язку й програмне забезпечення та дозволяє оператору керувати кількома БПЛА, в тому числі оперативно змінювати задачі.

У сучасних дослідженнях в галузі керування безпілотними апаратами багато дослідників сходяться на думці про необхідність застосування мультиагентних систем як наступний крок у розвитку цього напрямку. Наприклад, у [5] пропонується використовувати кооперативний підхід на основі мультиагентного навчання з підкріпленням, алгоритми якого враховують енергоспоживання та покриття території для оптимізації результатів.

Водночас в [6] пропонується запровадження подвійного режиму польоту, де для одних сценаріїв використовується мультиагентна система, а в інших – набір чітких правил для дій в особливо складних ситуаціях при невідомих середовищах.

Також пропонуються нові способи навчання скоординованої поведінки. Так у [7] представлений варіант методу багатоагентного глибокого детермінованого градієнта політики, який вирішує сценарії переслідування-ухилення від декількох цілей у нестаціонарних та невідомих середовищах з випадковими перешкодами.

Також важливим напрямком є комунікація між БПЛА, проблеми якої досліджуються в [8]. Зауважується, що штучно розроблені протоколи зв'язку зазвичай базуються на попередніх експертних знаннях і не мають гнучкості та адаптивності, що може обмежувати можливості взаємодії між БПЛА та не сприяє злагодженій роботі рою. В дослідженні визначена політика комунікації літальних апаратів, що дозволяє автономно визначати зміст повідомлення відповідно до свого поточного стану.

Автор [9] зауважує, що більшість існуючих методів штучного інтелекту для окремих агентів були розроблені для статичних середовищ, оскільки ними легше керувати та вони дозволяють проводити більш сувору математичну обробку. Наявність кількох агентів робить середовище динамічним, що може створювати проблему через нестабільну поведінку у випадку агентів, що навчаються одночасно.

Тож, задача керування зграями є доволі актуальною і викликає потребу в подальших дослідженнях.

1.2 Мультиагентний метод інтелектуалізації керування

Багатоагентні системи складаються з кількох агентів штучного інтелекту, які взаємодіють один з одним. Кожен агент працює незалежно, зі своїми власними цілями, можливостями та уявленням про середовище, проте робить свій внесок у загальну поведінку системи. Завдяки такій структурі багатоагентні системи здатні діяти у складних, розподілених або динамічних середовищах.

Розвиток інтелектуальних агентів як когнітивних систем пройшов через три ключові етапи, кожен з яких змінював розуміння поняття «інтелект» [10]. Класичний штучний інтелект підтримував концепцію символьних систем, де інтелект розглядався як здатність маніпулювати символами за логічними правилами. Когнітивна складова агентів була обмежена жорсткою логікою. Агентно-орієнтований підхід спирався на архітектуру «підпорядкування» зі створенням простого рефлекторного агента. Сучасний етап характеризується спробою об'єднати реактивність (швидкі рефлекси) з глибоким когнітивним аналізом [11]. Цей напрямок еволюціонував у створення «агентних фреймворків», де нейронні мережі виконують роль сенсорів та пам'яті, а стратегічне планування базується на архітектурах цілеспрямованих та навчальних агентів.

1.2.1 Основні компоненти когнітивної архітектури

Багатоагентні комп'ютеризовані системи мають кілька ключових компонентів [12]. Агент, як основний компонент мультиагентних систем, – це комплексна система, здатна сприймати своє середовище за допомогою датчиків та діяти за допомогою виконавчих механізмів [10], використовуючи штучний інтелект для прийняття рішень. Застосування багатьох агентів з розподіленим штучного інтелектом створюють мультиагентну систему, де агенти можуть взаємодіяти різними способами. При цьому агенти можуть бути гетерогенними [9], мати різну будову, функціонал та моделі знань.

Середовище, яке агенти здатні сприймати та в якому відбувається їх дія, може бути як реальне фізичне, так і штучно змодельоване. Зазвичай середовище є динамічним через вплив людини-оператора, взаємодію з іншими агентами, тощо.

Взаємодія між агентами включає співпрацю, координацію, обмін інформацією залежно від мети та типу агентів. Організація агентів відбувається через ієрархічний контроль з єдиного центру або через взаємодію елементів.

У класичному розумінні, запропонованому [10], інтелектуальний агент розглядається у циклі «Сприйняття – Дія» (рисунок 1.1).

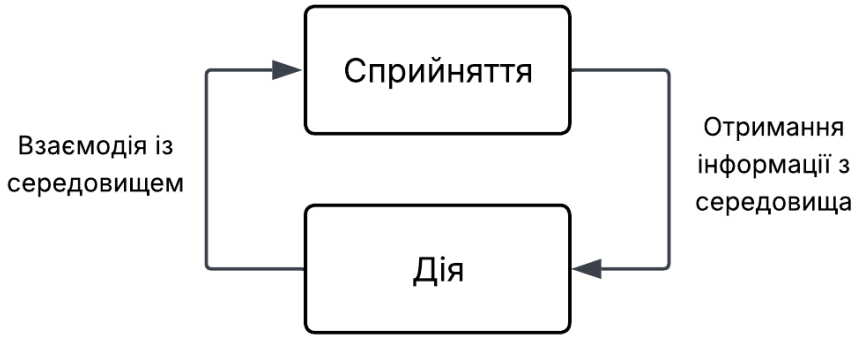


Рисунок 1.1 – Цикл «Сприйняття – Дія»

Проте сучасні дослідження структури «штучного розуму», зокрема [13], трактують агента як повноцінну когнітивну архітектуру. Когнітивна система функціонує в безперервному циклі, який називається «Standard Model of the Mind». Когнітивний цикл «Сприйняття – Міркування – Дія» зображено на рисунку 1.2.

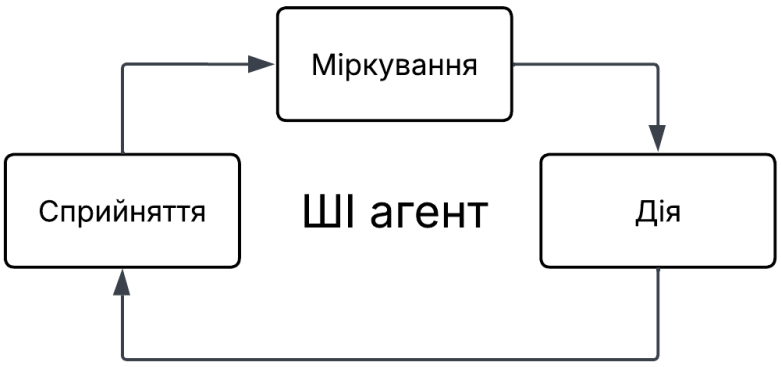


Рисунок 1.2 – Когнітивний цикл «Сприйняття – Міркування – Дія»

Як когнітивна система агент має:

– елементи сприйняття з перетворенням даних від сенсорів;

– пам’ять для зберігання та накопичення знань та алгоритмів роботи;

– механізм міркування.

Для раціональних агентів також передбачається наявність механізму модифікації через навчання на основі досвіду, а також механізм оцінки наслідків прийнятих рішень й зміни стратегії.

Механізм роботи агентного ШІ [14] схематично показаний на рисунку 1.3.

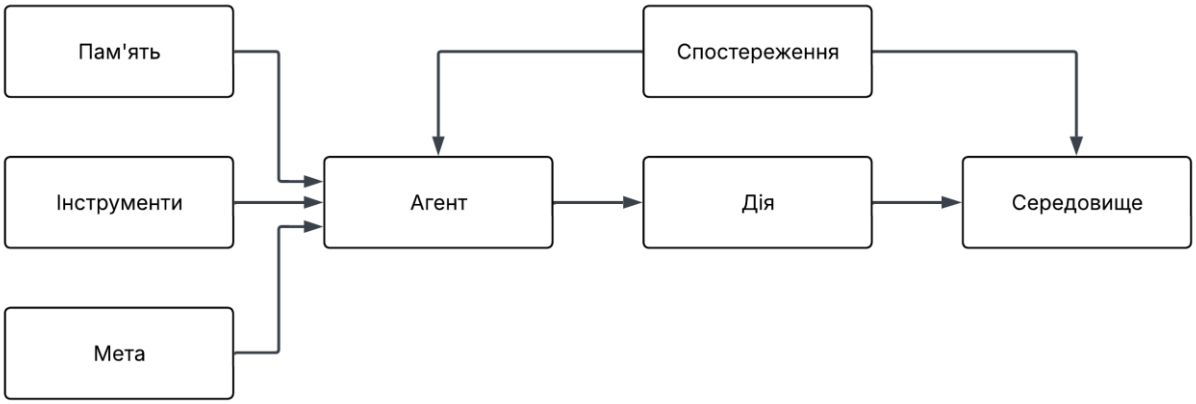


Рисунок 1.3 – Механізм роботи агентного ШІ

При сприйнятті відбувається фільтрування зовнішніх стимулів. Дані потрапляють у робочу пам’ять. Обробка інформації та прийняття рішень відбувається на основі знань. Реакція когнітивної системи спирається на підтримці актуальності внутрішнього стану та прогнозі майбутнього стану для визначення стратегії дій на основі знань про середовище. Інтелектуальні агенти працюють над досягненням визначеної мети, не потребуючи інструкцій на кожному кроці. Механізм міркування звертається до процедурної пам’яті, щоб знайти відповідне правило або дію. Може використовуватись інше ПЗ та джерела даних. Виконується дія, яка змінює середовище, і цикл починається знову.

1.2.2 Методи інтелектуалізації агентів

Агентний ШІ може включати комбінацію алгоритмів машинного навчання, великих мовних моделей та обробки природної мови, а також автономних процесів прийняття рішень та функціональності. Ці операції імітують людське пізнання, міркування та навчання.

Відповідно до обраної архітектури агенти можуть мати одну або декілька властивостей [15], наведених у таблиці 1.1.

Таблиця 1.1 – Властивості агентів

Властивість	Визначення
Автономність	Можливість діяти незалежно від користувача
Адаптивність	Здатність до навчання, зміни поводження на підставі досвіду
Комунікативність	Здатність до комунікації з користувачем або іншими агентами
Співробітництво	Взаємодія з іншими агентами для досягнення мети
Мобільність	Можливість переміщатися у навколишньому середовищі

З метою інтелектуалізації агентів застосовується ряд методів. Серед когнітивних методів, що дозволяють агенту вдосконалюватися на основі досвіду:

– машинне навчання: кероване, некероване (алгоритми кластеризації, зокрема K-means) та навчання з підкріпленням для можливості самостійного знаходження оптимальних рішень агентом;

– нейронні мережі та глибоке навчання для розпізнавання складних образів та прогнозування подій на основі великих даних.

Комунікативні методи за рахунок взаємодії в групі дозволяють інтелектуалізувати мультиагентну систему через створення протоколів

координації та кооперації для узгодження цілей між багатьма агентами для спільного розв’язання задачі. Інтеграція LLM дозволяє агенту розуміти природну мову, розбивати складні запити на підзадачі та викликати зовнішні інструменти (API).

Здатність мережі агентів автоматично змінювати свою структуру при зміні зовнішніх умов забезпечують методи самоорганізації системи.

Серед логіко-аналітичних методів, що використовуються для отримання нових знань з уже відомих фактів:

– баєсівські класифікатори для прийняття рішень в умовах невизначеності;

– нечітка логіка, що забезпечує здатність обробляти наближені дані та лінгвістичний формат мислення та використовувати експертний досвід;

– системи, засновані на знаннях, що реалізується через формальні описи понять та зв’язків між ними;

– дерева рішень для реалізації послідовних кроків для досягнення результату.

До методів інтелектуалізації агентів також відносяться групи методи адаптивного навчання, зокрема, еволюційні методи (генетичні алгоритми, ройовий інтелект), які шляхом стохастичної оптимізації виконують пошук найкращого рішення у величезному просторі варіантів, де звичайний перебір неможливий.

Мультиагентні системи мають ряд переваг перед традиційними підходами, зокрема [12]:

– гнучкість та масштабованість, адаптація до змін середовища шляхом модифікації агентів, що робить їх здатними для вирішення складних проблем;

– надійність та стійкість завдяки децентралізації керування, що дозволяє безперервно працювати навіть за умови виходу з ладу деяких компонентів;

– самоорганізація та координація на основі правил поведінки, розподілу завдання та скоординованого прийняття рішень;

– робота в режимі реального часу з негайним реагування на ситуацію без необхідності людського контролю, зокрема при виконанні пошукових чи рятувальних операцій, стихійних лих, тощо.

1.2.3 Сучасні напрямки розвитку інтелектуальних агентів

Огляд сучасних архітектур агентів представлено в [16]. Когнітивні агенти від жорстко заданих правил переходять до архітектур, заснованих на плануванні. Сучасні Goal-based agents використовують методи декомпозиції складних цілей на підзадачі. Пам’ять та планування реалізують нейронні мережі, використовуючи метод «ланцюжка думок» як частини когнітивного процесу. Використовуються векторні бази даних для імітації довготривалої пам’яті.

Метод Tree of Thoughts (дерево рішень) дозволяє агенту розглядати кілька варіантів розвитку подій одночасно з оцінкою кожного шляху, перш ніж зробити крок [17].

Дослідження [18] демонструють направленість на об’єднання класичних архітектур агентів із великими мовними моделями (LLM). Розвиток когнітивних агентів дозволив створювати LLM-based Agents. Замість жорстко прописаних правил агент використовує текстове міркування. Ключовий підхід ReAct (Reason + Act) – агент генерує текстове міркування і виконує дію [19].

Таким чином, дослідження провідних розробників спрямовані на переосмислення класичних архітектур з додаванням потужності нейронних мереж та інтеграції LLM моделей.

Компоненти сучасної когнітивної архітектури, що імітує структуру та механізми людського мислення для створення інтелектуальних систем, представлені на рисунку 1.4.

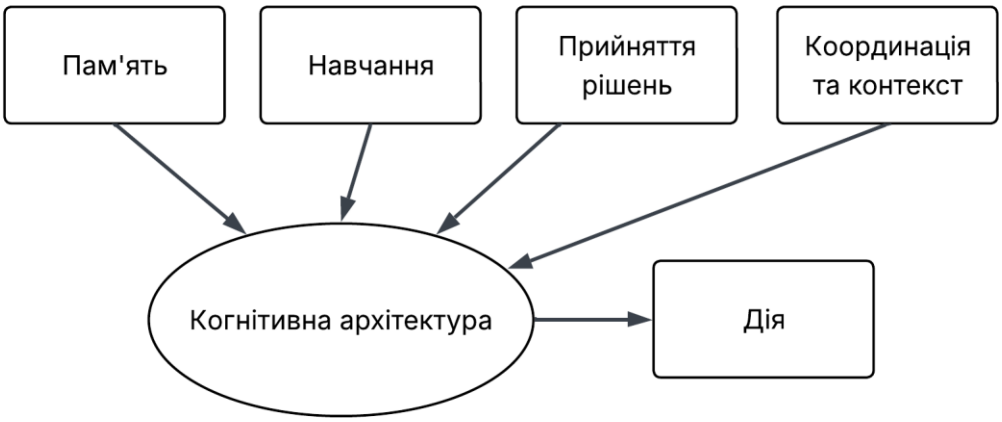


Рисунок 1.4 – Компоненти сучасної когнітивної архітектури

1.3 Задача керування зграєю як когнітивна модель

Як інтелектуальна система, що діє у відповідності до когнітивної моделі, мультиагентна система приймає рішення на основі аналізу поточних ситуацій для виконання поставленого завдання.

Функціональна схема роботи системи наведена на рисунку 1.5.

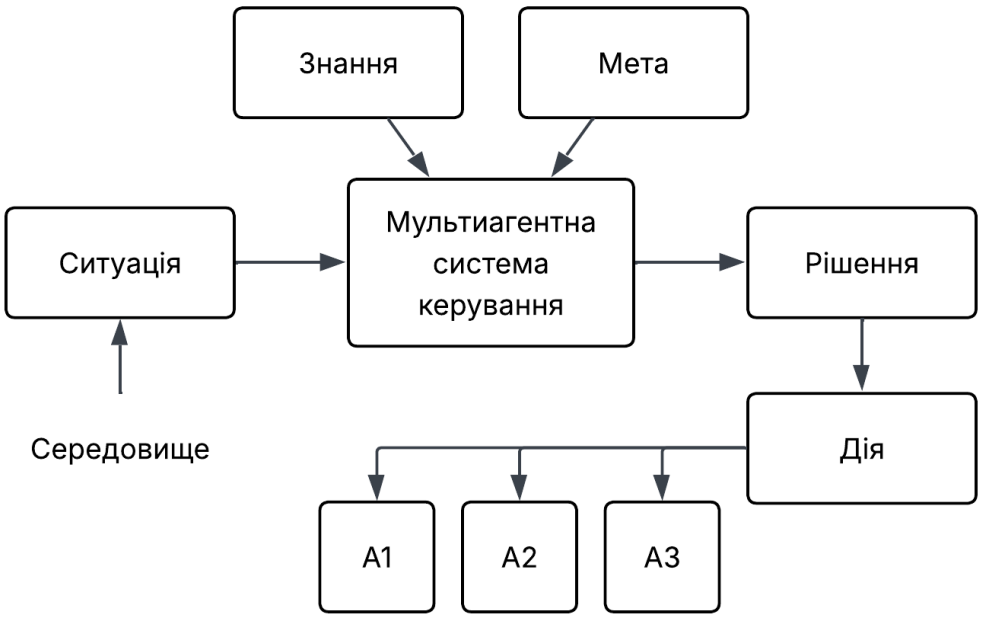


Рисунок 1.5 – Функціональна схема мультиагентної системи

Функціонування будь-якої інтелектуальної системи включає:

– сприйняття поточної ситуації;

– аналіз ситуації на основі бази знань системи;

– прийняття рішення, що узгоджується з поставленою метою;

– реалізація відповідного алгоритму реагування на ситуацію;

– протоколювання події та передача інформації на наземна станцію керування.

Застосування методів ройового інтелекту для керування дронами-зграєю в реальному часі передбачає побудову системи, що досліджуватиме визначену область зі збором та обробкою даних та збереженням результатів. Завдання, які планується вирішувати з використанням мультиагентної системи, є складними та багатоплановими з точки зору різнорідних задач, що мають вирішуватися одночасно в реальному часі. Кожен агент має індивідуальну мету і намагається її досягти в умовах динамічного середовища. Тому найбільша увага приділяється винайденню когнітивної моделі, що реалізується методами та засобами ройового інтелекту.

Мультиагентна система включає: агентів *(A)*, середовище *(E)* та зв’язки *(C)* [15], що формально може бути записано

$$M = \langle Аггенти, Середовище, Зв'язки \rangle. \quad (1.1)$$

Поняттям агента позначаються автономні компоненти, здатні до виконання певних функції та самостійного прийняття рішень з метою виконання індивідуального завдання для досягнення загальної мети. Кожен з агентів характеризується станом (*S*) та процесом (*P*), що виконується над даними й станом автономно $A_i = \langle S_i, P_i, I, O \rangle$. Стан агента визначає множина змінних $S = \langle x_1, x_2, ..., x_n \rangle$. Вхідні (*I*) й вихідні (*O*) змінні забезпечують зв’язок із середовищем. Середовище (*E*) включає процес і

стан $E_i = \langle S_i, P_i \rangle$, що змінюються незалежно від агентів. Когнітивна модель мультиагентної системи представлена на рисунку 1.6.

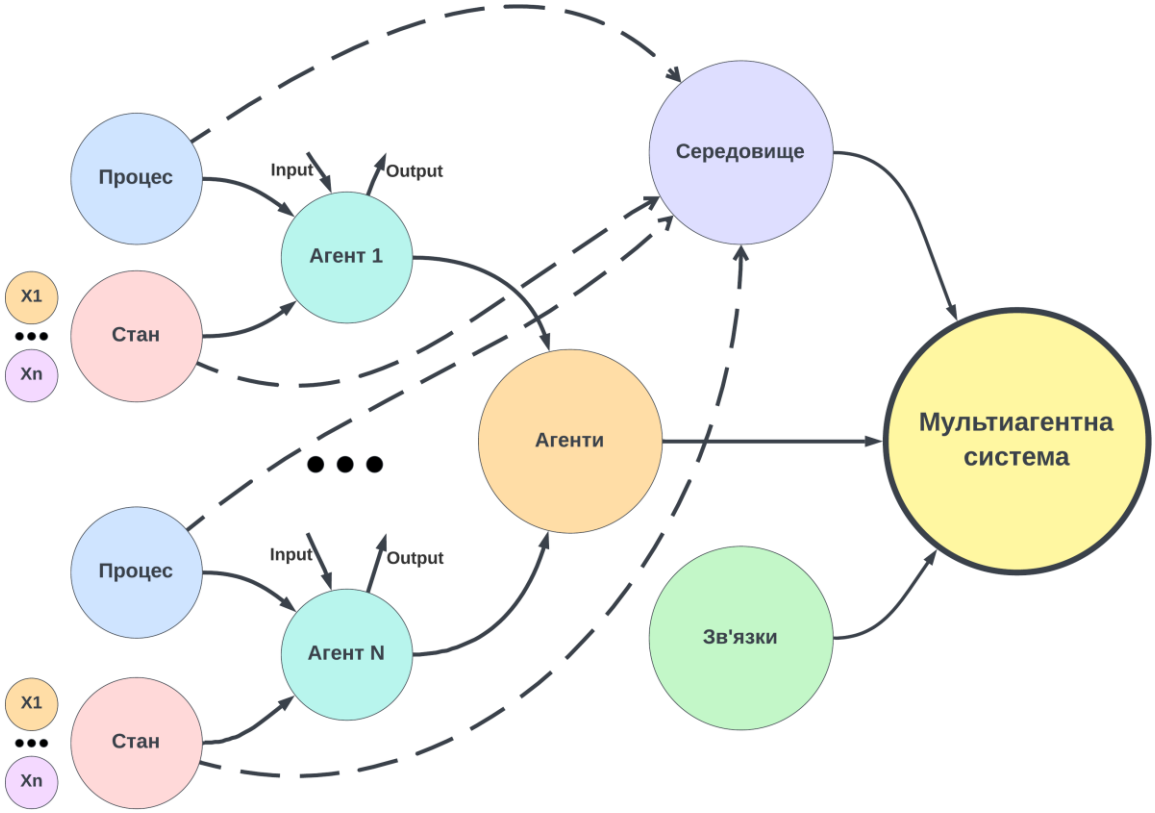


Рисунок 1.6 – Когнітивна модель

Мультиагентна система маючи внутрішню структуру, робочу та довготривалу пам’ять, здатна до моделювання навколишнього світу та активної взаємодії. Інтелектуальні агенти, що є практичною реалізацією когнітивних архітектур, об'єднують теоретичні моделі мислення людини: моделюють процеси сприйняття, навчання та планування, створюючи автономні адаптивні системи, здатні самосійно приймати рішення та досягати поставлених цілей.

Усвідомлення стратегії вирішення дозволяє вибрати оптимальні методи розв’язання в рамках мультиагентної технології та побудувати архітектуру системи з врахуванням специфіки завдання, що вирішуються.

1.4 Мета та постановка завдань дослідження

Сучасний етап розвитку інформаційних технологій зміщується від статичних алгоритмів до розвитку автономних інтелектуальних агентів, здатних функціонувати у динамічних середовищах. Застосування інтелектуальних агентів є однією з ключових тем в галузі AI у зв'язку з широким розповсюдженням інтелектуальних технологій і методів штучного інтелекту в усіх галузях. Когнітивна архітектура агентів імітує процеси людського мислення з метою досягнення автономності при прийнятті рішень та адаптивності до поточної ситуації. Автономні агенти використовуються в галузях, що потребують виконання масштабних завдань, при існуванні потенційного ризику для безпеки людини, зокрема, рятувальні операції, робота у віддалених та важкодоступних місцях.

Враховуючи актуальність як теми дослідження, так і використовуваної технології, метою роботи є: застосування методів ройового інтелекту для керування дронами-зграєю в реальному часі. Для цього потребують вирішити ряд завдань:

– огляд існуючих методів ройового інтелекту;
– огляд компонентів архітектури мультиагентної системи;
– огляд типів архітектур мультиагентних систем;
– розробка алгоритму роботи системи;
– розробка архітектури системи;
– вибір типів та розробка архітектури агентів;
– вибір мови та середовища програмування;
– програмна реалізація імітаційної моделі;
– оцінка отриманих результатів моделювання.

2 ОБҐРУНТУВАННЯ ВИБОРУ ТЕХНОЛОГІЧНИХ РІШЕНЬ

2.1 Методи та моделі ройового інтелекту

Дослідження груп автономних агентів, що рухаються, зосереджені на підтримці ефективності роботи, забезпеченні безперебійного зв’язку та координації із застосуванням методів ройового інтелекту. Дослідження підтвердили ефективність використання розподіленої архітектури з використанням оптимального розподілу завдань [20].

Незалежно від поставлених цілей та задач, керування групами автономних об’єктів потребує застосування ефективних алгоритмів координації руху та взаємодії незалежних агентів. Традиційні централізовані підходи мають ряд недоліків, зокрема недостатня ефективність, складність обчислень та вразливість до відмов. Натомість ройовий інтелект, як загальний метод вирішення складних задач через взаємодію багатьох простих агентів, вивчає колективну поведінку децентралізованих систем. В основі таких систем лежить принцип самоорганізації, коли складна глобальна поведінка виникає внаслідок локальних взаємодій між окремими елементами системи.

Розглянемо основні методи ройового інтелекту, що можуть застосовуватися для керування групами автономних агентів, зокрема безпілотних літальних апаратів [21], загальні принципи функціонування цих методів та застосування у задачах колективної координації та розподіленого управління.

2.1.1 Particle Swarm Optimization

Одним із найбільш відомих методів ройового інтелекту є стохастичний метод глобальної оптимізації Particle Swarm Optimization (PSO) [22], що був запропонований Джеймсом Кеннеді та

Расселом Еберхартом у 1995 році. Кожне потенційне рішення задачі представляється у вигляді частинки n_i, що переміщується у просторі пошуку R^n. Частинки обмінюються інформацією про найкращі знайдені положення та коригують свій рух відповідно до отриманого досвіду. Завдяки цьому система поступово збігається до оптимального або близького до оптимального розв'язку.

Формальне визначення простору пошуку: цільова функція, яку необхідно оптимізувати

$$f : R^n \to R_{\min} \tag{2.1}$$

Для пошуку в n-вимірному просторі рій складається з N частинок, де кожна в момент часу t характеризується двома векторами: позиція та швидкість. Поточне значення $x_i(t) = (x_{i1}, x_{i2}, ..., x_{in})$, вектор зміни позиції $v_i(t) = (v_{i1}, v_{i2}, ..., v_{in})$. Найкраща позиція, яку досягла і-частинка

$$p_i = \arg\min_{k \le t} \left[f\left(x_i(k)\right) \right], \tag{2.2}$$

а серед усіх частинок рою

$$g_i = \arg\min_{i \in \{1,...N\}} \left[f\left(p_i(t)\right) \right]. \tag{2.3}$$

Оновлення на кожному кроці алгоритму швидкості та позиції кожної частинки відбувається за формулами:

$$v_{ij}(t+1) = w \cdot v_{ij}(t) + c_1 \cdot r_1 \cdot \left(p_{ij}(t) - x_{ij}(t)\right) + c_2 \cdot r_2 \cdot \left(g_{ij}(t) - x_{ij}(t)\right) \tag{2.4}$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1)\Delta t. \tag{2.5}$$

Метод дозволяє реалізувати: планування траєкторію, огинання перешкод, розподіл завдань для окремих об’єктів, сумісний пошук, підтримка оптимального положення, налаштування параметрів, тощо.

2.1.2 Ant Colony Optimization

Поширеним підходом є алгоритм Ant Colony Optimization (ACO) [23], який моделює пошук найкоротших маршрутів між двома точками. ACO базується на ймовірнісному виборі шляху на графі. Реалізується шляхом використання числових параметрів, що характеризують інтенсивність на ребрах можливих рішень. У процесі ітерацій найбільш ефективні маршрути отримують більшу ймовірність їх подальшого використання.

Формально ймовірність переходу від *i*-точки до наступної j-точки визначається на основі значення інтенсивності τ на ребрі графа, видимості η, що є оберненою до довжини ребра, урахування досвіду і ефективності

$$P_{ij} = \frac{\tau_{ij}^{\alpha} \cdot \eta_{ij}^{\beta}}{\sum(\tau_{ij}^{\alpha} \cdot \eta_{ij}^{\beta})} \tag{2.6}$$

Оновлення значення інтенсивності на ребрах графу за формулою, що враховує коефіцієнт зменшення p (випаровування)

$$\tau_{ij}(t+1) = (1-p) \cdot \tau_{ij}(t) + \Delta\tau_{ij}. \tag{2.7}$$

Алгоритм ACO передбачає багато ітерацій, що заважає використанню для динамічного маневрування. Проте він ідеальний для попереднього планування. Тому цей метод найчастіше використовується не для прямого керування рухом, а для вирішення складних дискретних задач, таких як планування маршрутів та логістика: розподіл завдань між

об’єктами для оптимального обходу, знаходження оптимальної позиції для створення ланцюжка зв’язку, тощо.

2.1.3 Artificial Bee Colony

Потужним методом ройового інтелекту є алгоритм Artificial Bee Colony (ABC) [24], де агенти діють трьома підходами:

– оцінка кожного відомого варіанта рішення та локальний пошук

$$v_{ij} = x_{ij} + w_{ij}(x_{ij} - x_{kj}); \tag{2.8}$$

– вибір найбільш перспективного варіанта на основі отриманої інформації. Ймовірність P_i, пропорційна якості рішення ψ з точки зору наближення до цілі

$$P_{ij} = \frac{\psi_i}{\sum \psi_m}; \tag{2.9}$$

– генерація нового випадкового рішення, якщо протягом певного числа ітерацій рішення на покращується.

Вибір варіанта прямо пропорційний його якості, що робить алгоритм ABC більш стійким до різких коливань. Така стратегія дозволяє поєднувати інтенсивне дослідження перспективних рішень з глобальним пошуком.

2.1.4 Модель Boids

Серед моделей колективної поведінки окреме місце займає модель Boids [25], запропонована Крейгом Рейнольдсом у 1986 році. Кожен агент

керується трьома простими правилами, в результаті взаємодії яких виникає складна та реалістична колективна поведінка:

– уникнення зіткнень з сусідами;

– вирівнювання напрямку руху з локальною групою;

– рух до центру мас найближчих агентів.

Не маючи жорстких математичних структур, у Boids агенти переміщуються вільною групою, тримаючись поруч.

Формальний опис моделі: фінальний напрямок руху визначається шляхом постійного обчислення трьох векторів швидкості: розділення S, вирівнювання A, згуртованості C, а також вектора цілі T з коефіцієнтами w_i

$$V_{result} = w_1 S + w_2 A + w_3 C + w_4 T . \quad (2.10)$$

Застосування моделі ефективне при подоланні перешкод, слідуванні за агентом-лідером, створенні області спостереження, тощо.

2.2 Порівняльний аналіз типів архітектур агентів

Шлях розвитку штучного інтелекту від жорстко детермінованих систем до гнучких моделей, здатних до самонавчання відображає еволюція архітектур інтелектуальних агентів. В основі класифікації архітектур за рівнем внутрішньої складності лежить обсяг внутрішньої інформації про світ, яку агент використовує для прийняття рішень [10].

2.2.1 Прості рефлекторні агенти (Simple Reflex Agents)

Це системи «без пам’яті», які діють на основі поточного сприйняття та реалізації прямого зв’язку між сприйняттям та дією, ігноруючи попередню історію взаємодії. Механізм роботи базується на правилах типу «*умова – дія*». Такі агенти можуть працювати лише в умовах повного

спостереження за середовищем, що є обмеженням їх застосування. Якщо частина даних прихована, агент втрачає ефективність.

Математична модель простого рефлекторного агента базується на функціональному відображенні поточного сенсорного входу в дію. Функція логіки роботи агента може бути записана виразом

$$Action = f(Percept), \tag{2.11}$$

що реалізується через продукційну систему правил виду

$$IF\,(state \approx pattern)\,THEN\ action, \tag{2.12}$$

де *state* – поточна інтерпретація вхідних даних від сенсорів.

Такий агент не має внутрішнього стану (пам’яті). Схема агента з прямим потоком інформації без петель пам’яті на рисунку 2.1 [10].

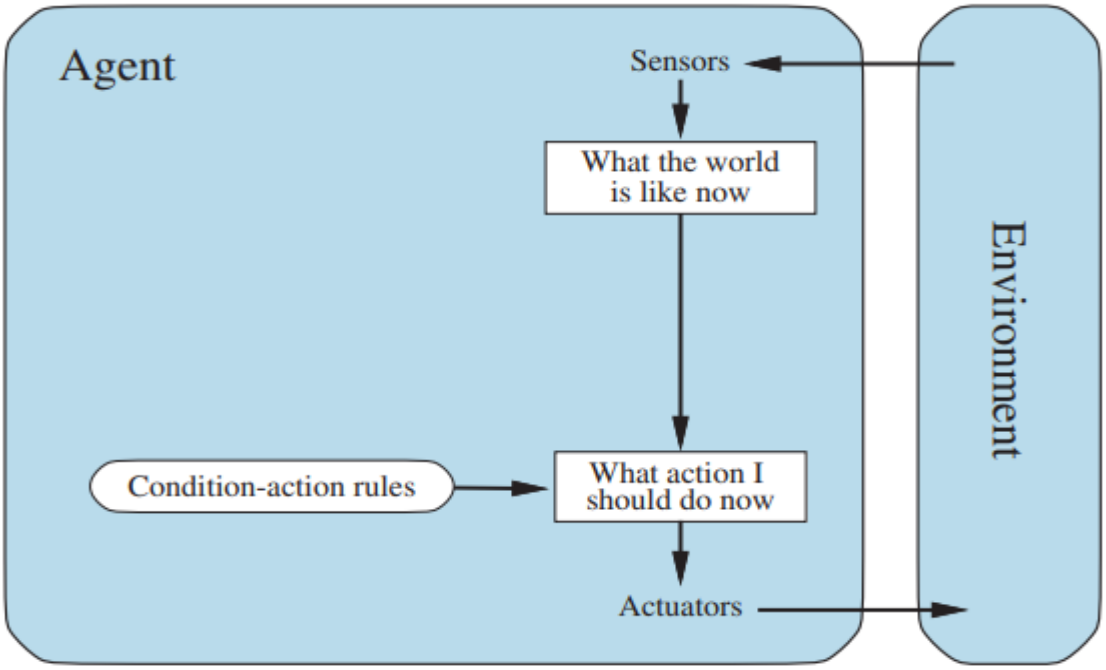


Рисунок 2.1 – Схема простого рефлекторного агента

Схема роботи агента. На основі інформації від сенсорів визначається стан середовища, За правилами «Condition → action» визначається, яку дію потрібно виконувати, та відбуваєтьсяії виконання.

2.2.2 Рефлекторні агенти на основі моделей (Model-based Reflex Agents)

Архітектура агента вдосконалена шляхом додавання внутрішнього стану для подолання обмеження повної спостережуваності. Внутрішня модель світу зберігає інформацію про ті частини середовища, які він не бачить у даний момент.

На кожному кроці t агент оновлює своє уявлення про світ за формулою:

$$S_t = Update\left(S_{t-1}, Action_{t-1}, Percept_t\right), \tag{2.13}$$

де S_t – поточний внутрішній стан;

S_{t-1} – попередній внутрішній стан;

$Action_{t-1}$ – остання дія агента, яка змінює середовище;

$Percept_t$ – поточне сприйняття нових даних від сенсорів;

Update – функція оновлення за алгоритмом, що враховує попередню. інформацію, власну дію та нові дані, формуючи актуальну картину світу.

Функція Update містить два типи знань:

– як світ змінюється незалежно від агента;

– як дії самого агента впливають на світ.

Схема агента на основі моделі (Model-based Reflex Agent) наведена на рисунку 2.2 [10].

Інтелект агента базується на здатності підтримувати внутрішній стан середовища та є стійким до тимчасових втрат сенсорних даних.

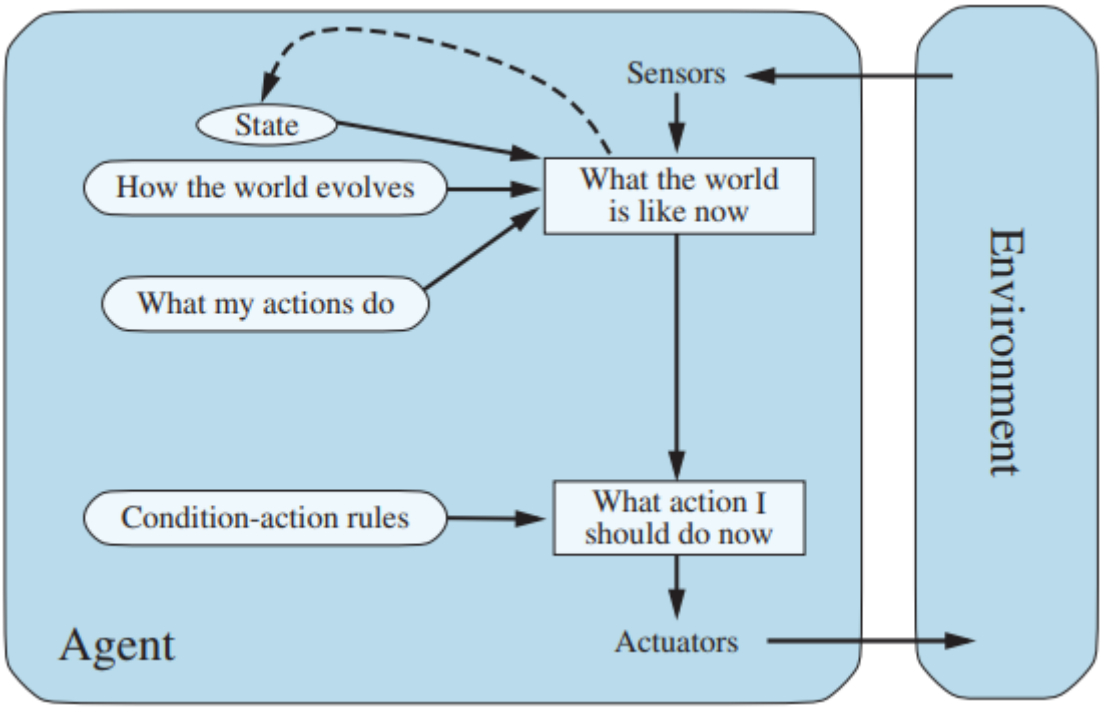


Рисунок 2.2 – Схема агента на основі моделі

Схема агента вміщує: блок стану, де зберігається опис поточного прихованого світу, модель динаміки середовища та модель впливу агента на зовнішній світ. Агент оновлює внутрішню картину світу перед тим, як звернутися до правил.

2.2.3 Агенти на основі цілей (Goal-based Agents)

Знання про поточний стан середовища не завжди достатньо для прийняття рішення. На відміну від рефлекторних, агенти на основі цілей діють, маючи певну мету, що описує бажані ситуації. В механізмі роботи використовуються алгоритми пошуку та планування на основі цілей. Вони оцінюють не лише поточний стан, а й те, чи наблизить конкретна дія до фінального результату. Для цього використовується складна внутрішня модель, що віщує: модель світу, оцінку станів та планування кроків шляху.

Для досягнення мети агенти мають знайти шлях у просторі станів від поточного S_{start} до цільового S_{goal} стану. Для цього агентом моделюються майбутні стани з прогнозуванням того, що буде, якщо виконати дію. потрібно обрати таку послідовність дій, що призведе до мети.

Перевагами такого підходу є гнучкість та адаптивність. Зміна цілі не потребує переписування всієї логіки, достатньо оновити опис мети. Агент автоматично перераховує маршрут, не потребуючи зміни базових правил поведінки.

Тож, блок прогнозування результатів є особливістю схеми агента на основі цілей, що зображена на рисунку 2.3 [10].

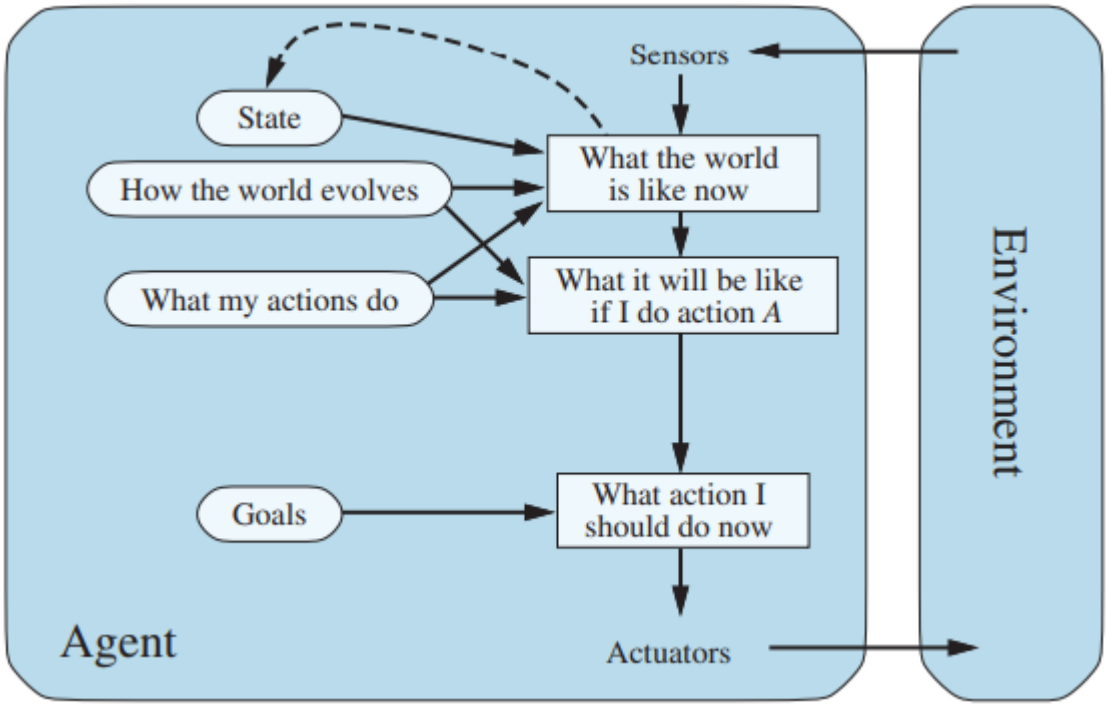


Рисунок 2.3 – Схема агента на основі цілей

Центральним елементом Goal-based Agents є блок цілей. Агент аналізує, що буде, якщо виконати певну дію, чи приведе результат до мети. Це вимагає залучення алгоритмів пошуку та планування.

2.2.4 Агенти на основі корисності (Utility-based Agents)

Цілі для таких агентів є бінарними (досягнуто/не досягнуто). Проте є багато шляхів до мети, деякі з яких швидші, безпечніші або дешевші. Агент використовує математичну функцію корисності для оцінки ефективності кожного стану. Це дозволяє приймати раціональні рішення в умовах конфлікту цілей (наприклад, швидкість проти економії палива).

Агенти на основі корисності використовують складніший математичний апарат – теорію раціонального вибору. Агент обирає дію, що максимізує очікувану корисність:

$$EU(a \mid e) = \sum_{S'} P(\mathrm{Re}\,sult(a) = S' \mid a, e) \cdot U(S'), \qquad (2.12)$$

де $P(\mathrm{Re}\,sult(a) = S' \mid a, e)$ – ймовірність того, що дія (a) призведе до стану (S') при наявних доказах (e);

$U(S')$ – функція корисності отриманого стану.

Застосування ці агенти знайшли при оптимізації шляхів, витрат палива, часу, тощо.

2.2.5 Навчальні агенти (Learning Agents)

Це найбільш просунутий тип когнітивних систем, здатний працювати в незнайомих середовищах. Він розглядається як ключовий атрибут інтелекту. Використовуючи навчання з учителем, навчання без учителя або навчання з підкріпленням, вони стають кращими в управлінні невизначеністю, узагальненні минулого досвіду та реагуванні на зміни навколишнього середовища.

Структура навчального агенту розділена на 4 концептуальні блоки:

– навчальний елемент відповідає за вдосконалення, вносить зміни в роботу виконавчого елемента на основі досвіду;

– виконавчий елемент – те, що раніше вважалося цілим агентом прийняття рішення. Він обирає зовнішні дії;

– критик оцінює агента згідно із зовнішнім стандартом, надає зворотний зв’язок щодо успішності дій;

– генератор проблем пропонує експериментальні дії, які можуть призвести до нового досвіду, виявлення нових ефективних стратегій.

Архітектура навчального агента полягає у взаємодії між виконавчим елементом та критиком. Критик обчислює сигнал помилки (*error*) у навчанні з підкріпленням

$$\delta_t = r_{t+1} + \gamma V(S_{t+1}) - V(S_t), \tag{2.13}$$

де r – винагорода,

V – оціночна вартість стану,

γ – коефіцієнт дисконтування.

Навчальний елемент використовує δ_t для оновлення стратегії виконавчого елемента, щоб у майбутньому обирати кращі дії. Генератор проблем впроваджує елемент дослідження (Exploration), змушуючи агента відхилятися від оптимальної стратегії для пошуку ще кращих рішень.

Схема навчального агента, наведена на рисунку 2.4, є найбільш розгалуженою [10]. Як видно зі схеми, агент структурно розділений на дві сфери: виконавчу та когнітивну. Когнітивна частина агента відповідає за сприйняття, мислення, навчання та прийняття рішень, тоді як виконавча відповідає за перетворення абстрактних рішень концептуального рівня в реальні дії, спрямовані на середовище. Окрім передача команд до виконавчих механізмів підтримується зворотний зв’язок з контролем виконання й фіксацією реакції середовища.

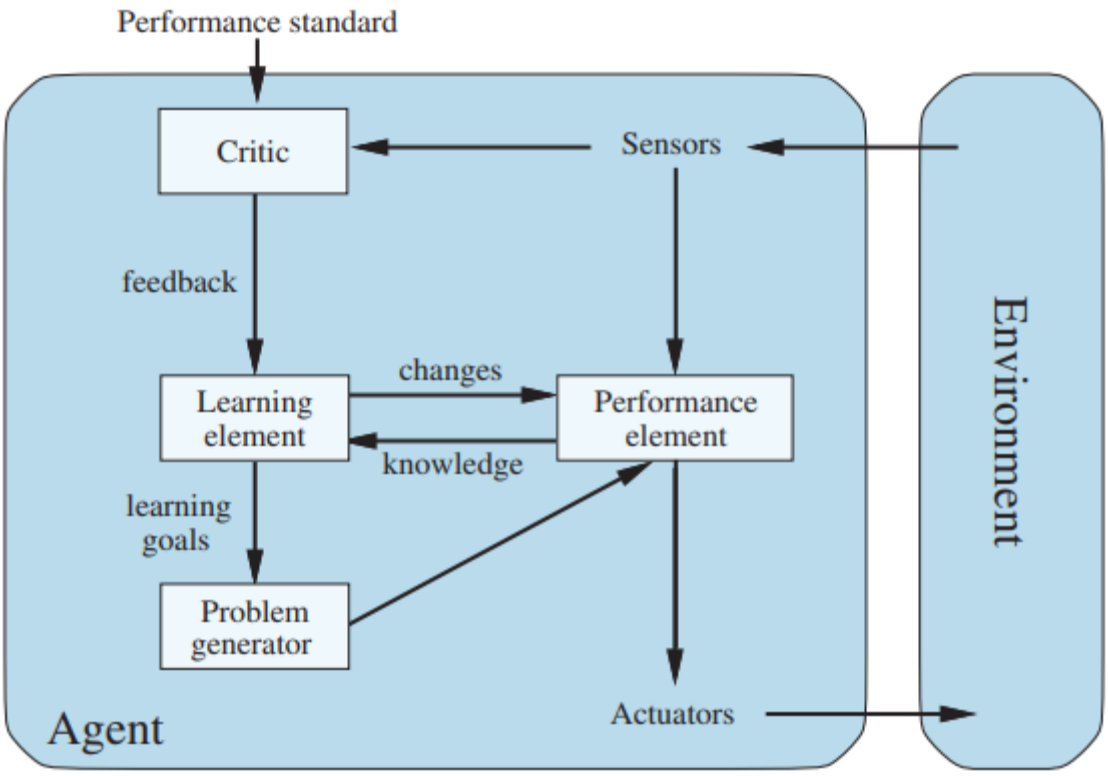


Рисунок 2.4 – Схема навчального агента

На основі огляду еволюції агентів від простих рефлекторних до навчальних демонструє перехід від жорсткого програмування до автономної поведінки. Якщо рефлекторний агент виконує лише запрограмовані дії, то навчальний агент здатний діяти в середовищах, про які немає повної інформації.

2.2.6 Порівняльний аналіз архітектурних підходів

Кожна описана архітектура інтелектуального агента має своє середовище застосування з набором умов, де вона є найбільш ефективною. В роботі [26] досліджено переваги та ризики навчальних агентів у непередбачуваних середовищах. Технічний розбір недоліків навчальних агентів, зокрема проблеми збіжності алгоритмів представлено в [10], а також в [27] наведено класичний опис переваг і недоліків.

Прості рефлекторні агенти (Simple Reflex), що працюють за принципом «стимул – реакція» мають ряд переваг:

– швидкість реакції з миттєвою відповіддю на вхідні дані, мінімальні обчислювальні витрати;

– низькі вимоги, адже вони не потребують потужних обчислювальних ресурсів чи великих обсягів пам’яті.

До суттєвих недоліків можна віднести: нездатність до навчання, обмеженість поведінки, відсутність адаптації до змін, окрім описаних в правилах ситуацій. Також в частково спостережуваних середовищах існує можливість зациклювання з повторюванням помилкових дій. Попри це рефлекторні агенти знайшли застосування в системах «розумного будинку», промислових роботах, ігрових NPC базового рівня.

Агенти на основі цілей (Goal-based) мають здатність до планування, а вибір дій базується на прогнозуванні майбутніх станів. Перевагами є:

– гнучкість, адже для зміни поведінки достатньо змінити опис цілі, а не всю логіку;

– здатність вирішувати складні задачі планування завдяки стратегічному міркуванню з можливістю будувати довгі ланцюжки дій для досягнення результату.

Недоліки: потребує значних ресурсів для прорахунку варіантів. Обчислювальна складність внаслідок експоненціального зростання складності пошуку в просторі станів. Має затримку через витрати часу на планування перед дією.

Навчальні агенти (Learning Agents**)** мають виняткову особливість – наявність механізму самовдосконалення через критику та досвід. Перевагами таких агентів є:

– автономність, можливість працювати в абсолютно невідомих середовищах;

– здатність до постійного самовдосконалення, завдяки чому зростає ефективність агента.

Недоліки: складність розробки, непередбачуваність поведінки або навіть небезпечність на етапі навчання. Агент може витрачати забагато часу на вивчення середовища. Попри це, навчальні агенти знайшли застосування в системах автопілотів, рекомендаційних системах та адаптивних системах.

Порівняльні характеристики за основними критеріями наведені в таблиці 2.1.

Таблиця 2.1 – Порівняльні характеристики агентів

Критерій	Simple agent	Goal-based	Leaning agent
Інтелект, когнітивна складність	низький (статичний)	середній (планування)	високий (адаптивний)
Знання, пам’ять	поточні правила	модель стану, мета	зворотний зв’язок, досвід
Гнучкість, адаптивність	немає	висока	максимальна
Складність	мінімальна	висока	дуже висока
Швидкість реакції	дуже висока	залежна від планування	висока після навчання

2.3 Аналіз сценарію вирішення задачі керування зграєю

Для вирішення завдання автономної роботи зграї БПЛА по дослідженню визначеної території з метою виявлення небезпечних предметів та адаптивне коригування завдань під час роботи необхідне визначення алгоритму взаємодії в багатоагентній системі.

На основі обробки інформації з бортової камери та сенсорного обладнання в реальному часі, має прийматися рішення про корегування

маршруту без участі людини-оператора. При цьому має бути висока достовірність результату як по виявленим об’єктам, так і по їх відсутності для недопущення хибної безпеки. Для цього місця виявлення підозрілих об’єктів мають бути додатково обстежені, для чого агент повинен додатково провести дослідження в цій зоні.

Складність завдання зростає пропорційно збільшенню площі і кількості автономних об’єктів, що співпрацюють над вирішенням поставленого завдання. Для цього можуть застосовуватись різні методи планування траєкторії для кількох БПЛА на основі навігації та розподілу завдань між агентами [28]. Алгоритми роботи мають знаходити оптимальні шляхи руху кожного безпілотника для повного покриття визначеної території при раціональному розподілі робочого простору між агентами.

Тож, основними завданнями керування є реалізація розподілу території з визначенням шляхів руху кожного БПЛА та забезпечення стабільного зв’язку як між агентами, так і з наземною станцією керування. Для виконання завдання мультиагентна інтелектуальна система керування має орієнтуватися в зовнішньому просторі, аналізувати поточні ситуації та адаптуватися коригуючи завдання.

Для цього потрібно розробити:

– архітектуру мультиагентної системи,

– типи агентів для розподілу завдань, виконання пошукових робіт та забезпечення зв’язку,

– архітектуру кожного типу агентів,

– програмне забезпечення для реалізації взаємодії агентів в мультиагентній системі.

Усвідомлення стратегії вирішення дозволяє вибрати оптимальні методи та засоби розв’язання в рамках мультиагентної технології та побудувати архітектуру системи з врахуванням специфіки завдання, що вирішуються.

2.4 Вибір технологічних рішень

Надійність системи напряму залежить від її складності – чим більше компонентів, тим більше точок потенційної відмови [4]. Для багатоагентної системи це означає, що ефективність залежить від балансу між складністю системи і її стійкістю до реальних умов. Тож, це має бути враховане при створенні системи керування.

Побудова багатоагентної системи, що досліджуватиме визначену область зі збором та обробкою даних та збереженням результатів передбачає застосування методів ройового інтелекту для керування дронами-зграєю в реальному часі. З точки зору різноманітності завдань агентів, що мають вирішуватися одночасно в реальному часі, завдання є складним та багатоплановим Кожен агент має індивідуальну мету, яку намагається досягти в умовах динамічно змінюваного середовища з яким взаємодіє через сенсори.

Типи агентів для виконання місії:

– агент-пошуковець: пошук за визначеним алгоритмом, передача даних. Автоматичне реагування на виявлені підозрілі предмети та ретельне обстеження небезпечної зони;

– агент-зв’язківець: оптимізація розміщення із забезпеченням безперервності зв’язку під час роботи;

– агент-координатор: розподіл та оптимізація завдання, корекція в динамічному режимі.

Для роботи з декількома різними типами агентів система має одночасно обробляти дані з різних джерел та координувати роботу агентів з використанням засобів зв’язку. Рух агентів відповідає моделі Boids для уникнення зіткнень.

3 МЕТОДИ ТА АЛГОРИТМИ РОЗВ'ЯЗАННЯ ЗАДАЧІ КЕРУВАННЯ ЗГРАЄЮ

3.1 Архітектура агентів

Дослідженням архітектур дронів впродовж останніх років присвячено багато робіт, зокрема [1], [3], [28], [29], де розглядалось апаратне оснащення як окремих дронів, так і у складі зграї.

В даному дослідженні для виконання моніторингу території з метою пошуку небезпечних об'єктів обрано багатоагентну архітектуру системи, що забезпечить оптимальне функціонування. Для організації прийняття рішень та управління застосована ієрархічна організація взаємодії агентів на кількох рівнях.

На вищому рівні людина-оператор планує та дає завдання на виконання процесу, адміністративний агент проміжного рівня розподіляє завдання та керує координацією, функціональні агенти нижчого рівня виконують завдання. Завдяки розподіленню завдання та розділенню функцій спрощується масштабування при розширенні системи, укрупненні завдання чи додаванні нових функцій мультиагентній системі.

3.1.1 Агент-пошуковець

Автономний ШІ розширює можливості агентів, інтелектуалізуючи прийняття рішень при виконанні поставлених завдань без необхідності втручання людини-оператора.

Розпізнавання зображень є одним з основних завдань пошукового агента при дослідженні території. Всі безпілотні летальні апарати оснащені бортовими комп'ютерами з відповідним програмним забезпеченням. Для візуального спостереження в мультиагентній системі застосовується комп'ютерний зір, а агентний штучний інтелект здійснює ідентифікацію

об’єктів за зображенням. Для цього агент обладнується навченою нейронною мережею на основі моделі класифікації для прийняття рішень щодо інтерпретації інформації з бортового обладнання, вибору й виконання відповідних дій при виявленні небезпечних об’єктів.

Класифікація вхідних даних здійснюється за визначеними класами небезпечних об’єктів. Агентний штучний інтелект у вигляді навчених моделей є когнітивним механізмом обробки вхідних даних і прийняття рішень [30], замінюючи людину-оператора. Такі моделі здатні інтегрувати інформацію з різних джерел – відеоряд, числові та сенсорні дані та вибирати відповідну подальшу поведінку.

Розпізнавання зображень використовується в багатьох сферах, тому існує багато технологій обробки зображень. Широке застосування отримав метод глибокого навчання нейронних мереж для виявлення та класифікації об’єктів на основі згорткових нейронних мережах. Згорткові нейронні мережі (CNN або ConvNet) [31] найбільш широко застосовуються для аналізу візуальних образів.

Базова архітектура мережі має дві основні частини, зображені на рисунку 3.1 [32].

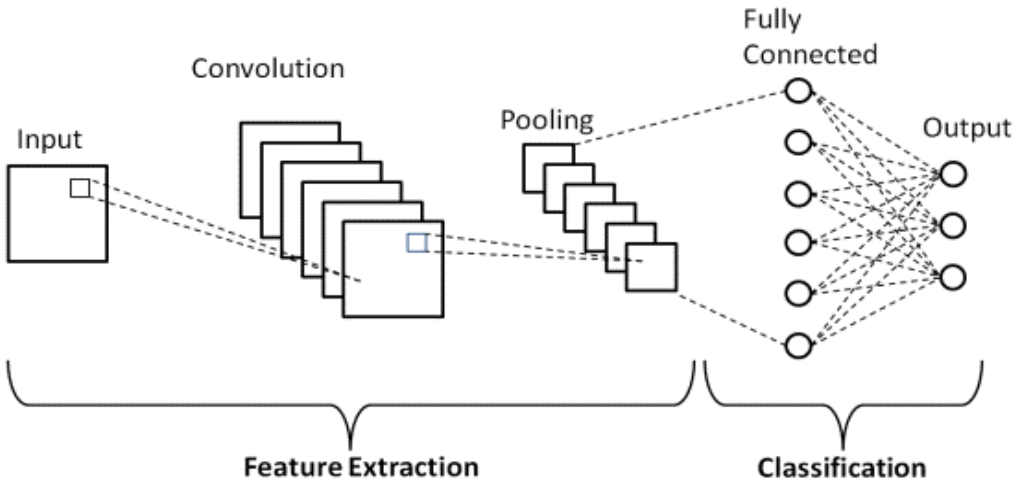


Рисунок 3.1 – Приклад багатошарової згорткової нейронної мережі для обробки зображень

Модель вилучення ознак CNN зменшує кількість ознак, присутніх у наборі даних через застосування математичної операції згортки, що розділяє та ідентифікує різні характеристики зображення. Класифікація зображення виконується повнозв'язним рівнем, використовуючи вихідні дані згорткової частини на основі ознак, виділених в попередніх шарах мережі. Розпізнавання об'єктів відбувається на основі виявлення кожного об'єкта на зображенні, його унікальних особливостей, концентрація на кожній важливій деталі незалежно від впливу зовнішніх чинників: розташування, освітлення, ракурсу огляду.

Одна з найперших і найпростіших – архітектура LeNet-5 CNN приведена на рисунку 3.2 [31].

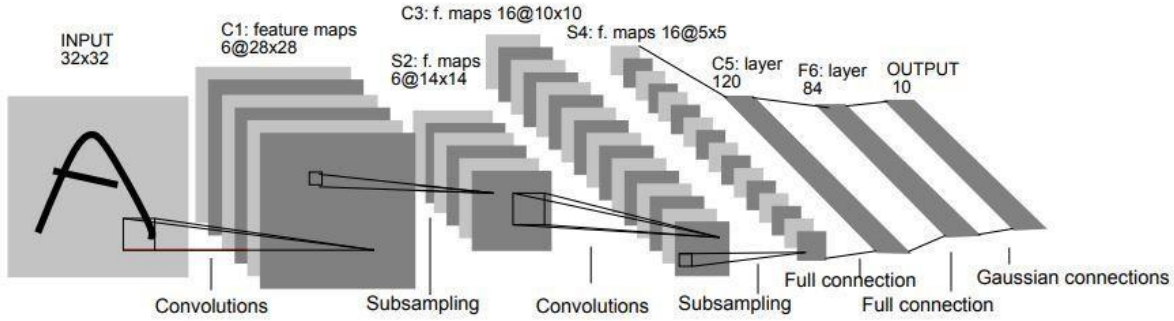


Рисунок 3.2 – Архітектура LeNet-5 CNN, представлена 7 рівнями

Впродовж наступних років було створено багато архітектур CNN [33], зокрема з адаптивною швидкістю навчання, збільшенням кількості рівнів, зменшенням помилок та спрощеням початкових модулів.

Таким чином, згорткові нейронні мережі є потужним інструментом для розпізнавання зображень у системах комп'ютерного зору, однак мережі цього класу можуть бути обмеженими для використання у мобільних моделях реального часу при автономній навігації. Залежно від роздільної здатності зображень, CNN можуть бути обчислювально витратними, що чинить вплив на швидкість розпізнавання об'єктів.

Алгоритм YOLO (You Only Look Once) порівняно з CNN є кращим вибором з точки зору оптимізації процесу розпізнавання вибухонебезпечних предметів, враховуючи особливі вимоги до швидкості обробки поточних даних. Він є оптимальним для роботи у моделях реального часу, при особливих вимогах до розпізнавання об’єктів у відеопотоці [34], [35]. Алгоритм YOLO використовує одночасну обробку всього зображення мережею для виявлення та класифікації об’єктів, що дозволяє підвищити швидкість для застосування при обмежених ресурсах. Сучасні алгоритми можуть бути впроваджені у безпілотних системах, роботах, що імітують когнітивні здатності людини.

З огляду на відсутність надійних універсальних засобів пошуку, що можуть виявляти різні типи вибухонебезпечних об’єктів на різній глибині залягання, в умовах щільної рослинності, архітектура моделі передбачає три рівні аналізу даних для підвищення достовірності виявлення:

– першим рівнем є модуль комп’ютерного зору, у якості якого використовується мікрокомп’ютер Nvidia Jetson Nano у поєднанні з алгоритмом виявлення об’єктів Ultralytics YOLOv8, що забезпечує швидке виявлення об’єктів у полі зору оптичної камери;

– другим є використання магнітометра для сканування поверхні;

– третім рівнем є тепловізійна камера у якості додаткового датчика.

Такий підхід дозволяє трикратно підтвердити виявлення об’єкта, що дає додаткову впевненість у точності результату.

Архітектура дрону, як пошукового агента, включає:

– автопілот,

– бортовий мікрокомп’ютер,

– приймач-передавач,

– GPS-навігатор,

– камера,

– металошукач;

– тепловізійний датчик.

Додавання периферійних пристроїв дозволяє розширювати функціональні можливості агентів у відповідності до цілей і виконуваних завдань та підтримки протоколів комунікації з іншими пристроями.

3.1.2 Агент-зв’язківець

Архітектура систем управління безпілотними летальними апаратами має свої обмеження. Основним фактором є залежність від стабільності та якості зв’язку – при його втраті управління може бути обмежене або повністю втрачене. Також важливу роль відіграють фізичні перешкоди, перевантаження каналів та обмеження пропускної здатності каналу зв’язку. Віддаленість від наземної стації чи велика кількість даних також можуть створювати затримки [3], [4].

Для забезпечення безперебійної роботи потрібен стабільний канал зв’язку, оскільки керування зграєю БПЛА базується на цілісній системі, що поєднує бортові пристрої між агентами та з наземною станцією керування. Це особливо важливо при нестабільному зв’язку у віддаленій місцевості, під впливом завад чи фізичних перешкод [28]. Тому для розширення зони управління зграєю використовуються дрони як ретранслятори даних. Вони приймають інформацію і передають її по ланцюжку функціональним агентам. Це дозволяє керувати на великій відстані від базової станції та при ускладнених умовах середовища.

На рисунку 3.3 показана схема забезпечення безперебійного зв’язку у багатоагентній системі за алгоритмом Ant Colony Optimization.

Передачу даних від агента-координатора щодо керування завданням забезпечує бортовий комп’ютер через агентів зв’язку. За алгоритмом мурахи агенти підтримують оптимальне місце розташування, забезпечуючи постійний зв’язок з усіма агентами, що входять в зграю. Це робить можливим виконання завдання навіть при складній формі ділянки, що обстежується.



Рисунок 3.3 – Схема забезпечення безперебійного зв’язку

Архітектура дрону, як агента-зв’язківця, включає:

– автопілот,

– бортовий мікрокомп’ютер,

– приймач-передавач,

– GPS-навігатор.

Організація каналів зв’язку є вкрай важливим елементом керування та координації БПЛА у зграї. Додавання периферійних пристроїв дозволяє розширювати функціональні можливості агентів у відповідності до цілей та виконуваних завдань для підтримки протоколів комунікації з іншими дронами зграї. Інтеграція додаткових датчиків дозволяє дронам ефективно працювати в складі рою з колективним збором інформації.

За необхідності є можливість застосування плат розширення, що дозволить вибирати види зв’язку [3]. Можливе також використання кількох каналів одночасно: для команд та відео, що сприяє підвищенню стабільності при частковій втраті сигналу [4] та забезпечує високу надійність.

3.1.3 Агент-координатор

Важливою складовою дослідження ділянки з використанням рою дронів-агентів є оптимізація розподілу завдання. Для цього використовується агент-координатор. Під час роботи агент-координатор отримує завдання від людини-оператора у вигляді вхідних даних. Його завданням є обчислення оптимального розподілу завдання між заданою кількістю агентів, координація дій і швидке реагування на зміну обстановки та, за необхідності, ситуативна оптимізація та перерозподіл завдань під час роботи. Маршрутні завдання від агента-координатора надсилаються до бортового комп'ютера кожного функціонального агента, що виконують пошук.

Людина-оператор визначає межі пошукової області та задає кількість дронів для пошуку. Завдання координатора полягає в отриманні від користувача набору координат, що можуть задати довільну багатокутну форму досліджуваної території. Процес обробки вхідних даних складається з аналізу і сегментації простору та генерації індивідуальних траєкторій для кожного агента.

Алгоритм роботи агента-координатора наступний. Спочатку визначаються крайні точки багатокутника для формування віртуальної прямокутної області охоплення, використовуючи алгоритм пошуку мінімальних та максимальних значень координат змінних X_{min}, X_{max}, Y_{min} та Y_{max} з подальшою ітерацією по всіх вершинах. Математично це виражається як знаходження значень

$$X_{\min} = \min(X_i), \tag{3.1}$$

$$X_{\max} = \max(X_i), \ i = 0,1,...,n, \tag{3.2}$$

де n – кількість кутів заданого багатокутника.

Це дозволяє локалізувати пошук всередині конкретного віртуального поля, що дає можливість оптимізувати подальші розрахунки.

Після визначення загальних меж території обстеження агент-координатор приступає до стратегічного розподілу простору між заданою кількістю агентів-пошуковців. Використовується вертикальна декомпозиція, де ширина w кожного індивідуального сектора розраховується за формулою

$$w = \frac{X_{\max} - X_{\min}}{q}, \tag{3.3}$$

де w – ширина індивідуального сектора;

q – кількість функціональних агентів.

Для кожного i-го дрона визначаються межі його персональної відповідальності:

– ліва межа

$$x_{i\min} = X_{\min} + w \cdot j, \tag{3.4}$$

– права межа

$$x_{i\max} = X_{\min} + w \cdot (j+1), \quad j = 1, 2, ..., q. \tag{3.5}$$

Таким чином, вся територія розділяється на рівні ділянки, що гарантує відсутність перекриття призначених агентам завдань та забезпечується повне покриття заданої території. Кінцевим результатом роботи агента-координатора є генерація списку координат індивідуального сектора для кожного дрону. За потреби може відбуватися корекція завдання з перерозподілом залишку території при неоднаковій завантаженості. Тоді вільний агент допомагає сусіднім.

3.1.4 Навігація агентів

При розробці системи припускається наявність групи з q агентів-пошуковців та k агентів-зв'язківців на території, що задається набором координат точок, визначених людиною-оператором. Всі БПЛА мають злетіти одночасно та дотримуватись безпечного руху до різних початкових позицій на індивідуальних секторах обстеження.

Рух агентів в секторі обстеження відбувається наступним чином. При русі дрона перевіряється належність точок всередині i-го сектора до фактично обраного користувачем полігону. Оскільки полігон може мати складну багатокутну форму, просте сканування прямокутного сектора призведе до виходу дрону за межі цільової зони. Для вирішення цієї проблеми використовується алгоритм перевірки перетину лінії обмеження сектора з ребрами заданого багатокутника для кожної точки (x, y).

Методом перевірки належності точки до багатокутника [36], [37] перевіряється виконанням умов

$$(y_{\min} > y_i) \neq (y_{\max} > y_i). \tag{3.6}$$

Це дозволяє перевірити, чи знаходиться горизонтальна лінія, проведена через точку (x_i,, y_i), між двома вершинами ребра: y_{min} та y_{max} визначеного сектора. Якщо одна точка вище лінії, а інша нижче, значить точка (x_i,, y_i) перетнула ребро сектора.

Водночас розраховується координата x точки перетину лінії та ребра з використанням умови

$$x_i < \frac{(x_{\max} - x_{\min}) \cdot (y_i - y_{\min})}{(y_{\max} - y_{\min}) + x_{\min}}. \tag{3.7}$$

Якщо координата x_i менша, значить лінія дійсно перетинає ребро сектора.

Виконання умов дозволяє в процесі роботи динамічно визначати, чи знаходиться конкретна координата в заданих межах сектора, що є основою для побудови точного маршруту. Якщо поточна точка відповідає наведеними умовами, вона входить до маршруту конкретного агента. Кожного разу, коли перетинається ребро сектора з виходом за межі, напрямок руху агента змінюється на протилежний.

3.2 Архітектура та алгоритм роботи мультиагентної системи

Архітектура багатоагентної системи керування зграєю Безпілотних летальних апаратів визначає організацію взаємодії між агентами, передачу інформації та її обробку. У ході дослідження було створено ієрархічну розподілену архітектуру з наземною станцією керування системою, представлену на рисунку 3.4.

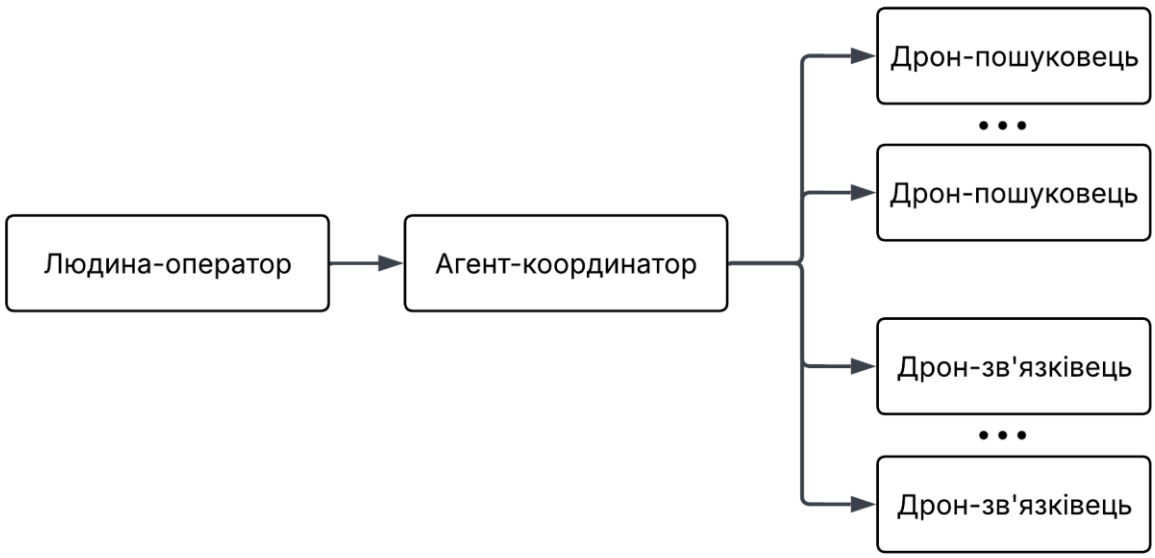


Рисунок 3.4 – Ієрархічна розподілена архітектура системи керування зграєю

Керування розподіленими агентами ґрунтується на співпраці між агентами та забезпеченні стабільного зв’язку для їх взаємодії. Для цього

архітектура агентів вміщує ряд елементів для забезпечення керування та взаємодії: бортовий комп’ютер, модулі зв’язку, GPS-навігатори.

Бортовий комп’ютер відповідає за обробку команд від агента-координатора, виконання завдання, взаємодію всіх елементів системи. Модулі зв’язку, що включають приймання та передачу даних, команд керування та результатів обстеження.

Центральним елементом є наземна станція керування, обладнана інтерфейсом для взаємодії з людиною-оператором, вводу даних та зберігання отриманих результатів. Передбачене вікно спостереження за ходом виконання завдання, що дозволяє бачити загальну ситуацію з відображенням роботи кожного дрона-агента зграї.

Призначення завдань для безпілотників відбувається оператором на комп’ютері, після чого агент-координатор розподіляє обрану ділянку між дронами та передає завдання для дронів-пошуковців через канал зв’язку, підтримуваний дронами-зв’язківцями, які в свою чергу, використовують методи мурашиного інтелекту для підтримки рівної відстані до всіх дронів.

БПЛА сканує територію в автономному режимі за розрахованим маршрутом. На бортовому мікрокомп’ютері постійно відбувається отримання даних з трьох датчиків (камера, магнітометр, тепловізор) та виявлення ознак наявності шуканих об’єктів.

Сканування індивідуального сектора пошуку для кожного дрона-агента відбувається по лінійній траєкторії човниково (рисунок 3.5). При цьому всі БПЛА мають рухатися на однаковій висоти та зберігати постійну відстань між смугами при русі.

Дрон з початкової точки в заданому секторі прямує до протилежної межі сектора, перевіряється умова досягнення ребра, після чого дрон зміщується на відповідну відстань до наступної смуги та летить у зворотному напрямку. При цьому частина поля зору при новому прольоті має частково перекривати територію, просканову при попередньому проході для подвійного контролю.

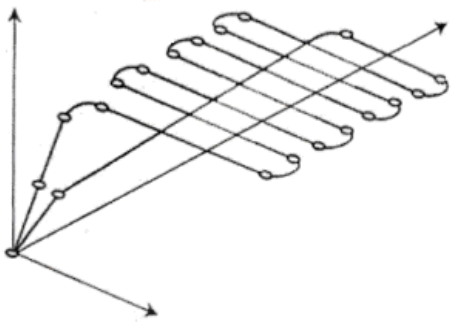


Рисунок 3.5 – Човниковий маршрут руху дрона-пошуковця

Зона перекриття, показана на рисунку 3.6, залежить від поля зору камери, та висоти польоту й може бути обчислена математично [28].

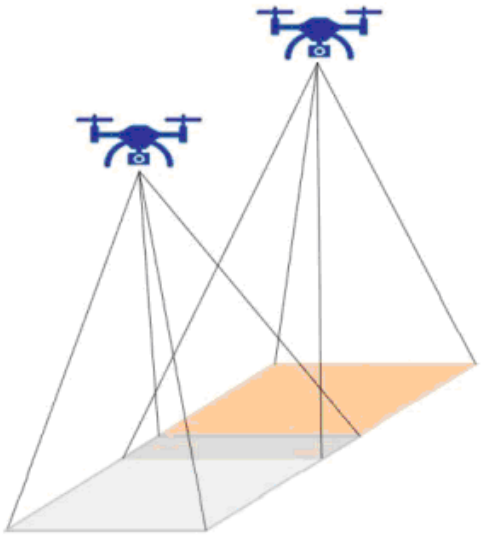


Рисунок 3.6 – Зона часткового перекриття

Після закінчення поздовжнього сканування всього сектору, дрон починає сканування сектора поперечно, що забезпечує повторне сканування з іншим кутом зору для отримання високої достовірності виявлення та запобігання пропуску небезпечних об’єктів.

При виявленні об’єкта будь-яким з датчиків, фіксується його місцезнаходження та заноситься до списку потенційно небезпечних.

Додатково здійснюється обліт довкола виявленого об’єкта для сканування під різними кутами зору. Якщо усі три датчики підтверджують наявність об’єкта, зображення з ним та його географічні координати заносяться на інтерактивну мапу, даючи змогу оператору переглядати результати сканування місцевості у реальному часі.

Під час виконання завдання зі сканування також важливо не допустити зіткнення літальних апаратів, що призвело б до втрати обладнання. Для цього передбачено й реалізовано поведінку ухилення, коли при зближенні двох дронів один з них набирає висоту, а після віддалення повертається на попередньо задану.

Реалізація системи забезпечує повний цикл робіт з дослідження визначеної території – від отримання завдання до протоколювання результатів обстеження, що виконуються автономно зграєю БПЛА-агентів. Кожен агент сприяє досягненню колективної мети шляхом виявлення й передачі нових даних на наземну станцію, з відтворенням на карті інформації в on-line режимі.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ СПЕЦІАЛІЗОВАНОЇ БАГАТОАГЕНТНОЇ СИСТЕМИ

4.1 Технології та інструменти

Вибір набору технологій для створення системи керування зграєю дронів зумовлений необхідністю забезпечення гнучкої розподіленої архітектури та можливості асинхронного обміну даними у реальному часі. Основною мовою програмування інтелектуального ядра було обрано Java, оскільки вона надає потужні засоби для роботи з потоками, об’єктно-орієнтованим моделюванням та мережевою взаємодією. Основою для побудови агентів було обрано спеціалізовану платформу Java Agent Development Framework, також відому як JADE. Цей фреймворк забезпечує готові абстракції, наприклад CyclicBehaviour та TickerBehaviour, для управління життєвим циклом агентів, реалізує стандарти комунікації FIPA (Foundation for Intelligent Physical Agents). Завдяки JADE кожен дрон моделюється як повністю автономна сутність, здатна самостійно приймати рішення, обробляти внутрішні стани та реагувати на повідомлення від інших агентів зграї.

Також було використано мову Python у поєднанні з бібліотекою Pygame, завдяки чому було реалізовано просторову тривимірну візуалізацію поточного стану системи у реальному часі. Бібліотека Pygame виконує роль багатофункціонального модуля візуалізації та інтерфейсу користувача, забезпечуючи інтерактивну взаємодію з інтелектуальним ядром системи. На етапі початку роботи графічне середовище функціонує в режимі графічного конфігуратора, що дозволяє людині-оператору вручну визначати межі робочої ділянки довільної геометричної форми. Програмна реалізація цього режиму базується на обробці подій маніпулятора, за допомогою якого формується масив координат вершини багатокутника,

що згодом передається агенту-координатору для виконання розрахунків секторів сканування.

4.2 Архітектура програмного забезпечення

Система побудована з використанням принципів багатопотоковості. Отримання зображень з камери та їх обробка нейронними мережами виконуються різними обчислювальними потоками.

Архітектура програмного забезпечення системи керування збудована за децентралізованим принципом з використанням парадигми агентно-орієнтованого програмування. Загальна інфраструктура розгорнута у головному контейнері середовища виконання JADE, де реєструються всі учасники симуляції. Базовим допоміжним вузлом виступає агент станції StationAgent, який бере на себе початкові завдання ініціалізації середовища: динамічну генерацію розмірів робочого поля в межах заданих параметрів та випадкове просторове розташування цілей. Після розсилання початкових параметрів, наземна станція переходить у режим очікування даних від агентів та паралельного транслювання загальної інформації для візуалізатора.

Рій безпілотних апаратів складається з множини ідентичних за своєю програмною структурою агентів-дронів, кожен з яких виконує власний цикл поведінки. Життєвий цикл такого агента базується на паралельному виконанні двох ключових процесів:

– циклічного прослуховування ефіру MessageReceiver для приймання вхідних повідомлень, наприклад, команд, координат сусідніх агентів;

– регулярних таймерних оновлень власного стану з частотою 10 Гц.

Усі компоненти програмного забезпечення функціонують асинхронно, що гарантує високу відмовостійкість системи. Замість використання єдиного центру мікрокерування кожен дрон спирається на внутрішній скінченний автомат (стани IDLE, SEARCHING, RETURNING)

та локальні правила взаємодії, самостійно плануючи свою траєкторію та координуючи дії з сусідніми агентами через обмін ACL-повідомленнями.

Візуалізація польотних даних у тривимірному просторі реалізована за допомогою математичних методів проектування координат на двовимірну площину екрана. Оскільки стандартні засоби бібліотеки орієнтовані на 2D-графіку, для відображення ешелонування дронів за висотою застосовано афінні перетворення та тригонометричні матриці повороту. Це дає змогу оператору динамічно змінювати кут огляду сцени та візуально контролювати вертикальне розведення апаратів у зонах потенційних колізій. Додатково реалізовано графічне відображення зон покриття у вигляді динамічної сітки, що дозволяє в реальному часі оцінювати ефективність обстеження території та цілісність сформованої мережі зв’язку.

Взаємодія між логічним ядром на базі Java та візуалізатором на базі Python організована через мережеві сокети за протоколом UDP (User Datagram Protocol). Використання датаграм (неблокуючих сокетів) дозволяє досягти мінімальної затримки під час передачі координат та статусів дронів [38], [39]. Такий архітектурний поділ на обчислювальне ядро та незалежний графічний візуалізатор гарантує, що процеси відображення графіки не будуть гальмувати або блокувати швидкі динамічні процеси прийняття рішень усередині системи агентів.

4.3 Програмна реалізація системи керування

Програмна реалізація системи керування з представленням інтелектуальних агентів у вигляді функціональних вузлів та автономних модулів, дозволяє реалізувати стратегії групової поведінки.

Програмний код структуровано у вигляді набору спеціалізованих класів, які успадковують базовий клас Agent з фреймворку JADE. Кожен

клас інкапсулює специфічну логіку поведінки, переходи між станами та обробку подій.

Особливістю створеної системи є використання методів ройового інтелекту у ітераційних циклах керування, що мають частоту оновлення 100 мс, гарантуючи детерміновану й плавну реакцію дронів на зміни навколишнього середовища та забезпечує стабільнеь функціонування.

Алгоритми просторового орієнтування та динамічного перерозподілу даних побудовані на принципах локальної взаємодії та самоорганізації агентів-дронів, що мінімізує залежність системи від центрального вузла управління.

Обчислення нових координат кожного БПЛА відбувається на основі поточного просторового положення та розрахованого вектора до цільової точки (x_{target}, y_{target}, z_{target}).

$$\begin{aligned} d_x &= x_{target} - x \\ d_y &= y_{target} - y, \\ d_z &= d_{target} - z \end{aligned} \tag{4.1}$$

$$dist = \sqrt{d_x^{\,2} + d_y^{\,2} + d_z^{\,2}}\,. \tag{4.2}$$

У процесі навігації агент самостійно нормалізує вектор напрямку руху та при перевищенні мінімального порогу дистанції *dist > 1* оновлює свої координати за правилом

$$\begin{cases} x_{new} = x + \dfrac{d_x}{dist} \cdot V \\ y_{new} = y + \dfrac{d_y}{dist} \cdot V \\ z_{new} = z + \dfrac{d_z}{dist} \cdot V \end{cases}, \tag{3.5}$$

де V – фіксована швидкість руху, що забезпечує рівномірне і передбачуване переміщення у тривимірному просторі.

4.3.1 Модуль навігації та уникнення зіткнень

Модуль навігації відповідає за просторове переміщення дрона й запобігання аварійним ситуаціям за допомогою механізмів самоорганізації. Отримавши межі сектору дослідження від агента-координатора, дрон самостійно визначає траєкторію руху, розраховуючи масив точок траєкторії для сканування території у човниковий спосіб. Алгоритм забезпечує рух дрона від мінімальної координати сектора x_{min} до максимальної x_{max}, після чого агент здійснює зміщення координати за віссю Y на фіксований крок з урахуванням перекриття і змінює напрямок руху по вісі X на протилежний. Такий підхід забезпечує автономне виконання операції сканування призначеного сектора без необхідності у постійному отриманні інструкцій від центрального вузла.

Під час кожного такту оновлення навігаційний модуль перевіряє поточну відстань до всіх відомих сусідніх дронів, останні координати яких безперервно оновлюються у локальній хеш-таблиці.

Для забезпечення безпеки польотів у тривимірному просторі реалізовано децентралізований алгоритм вертикального ешелонування за висотою для уникнення загроз зіткнення за принципом розділення ройової моделі Boids. Якщо розрахована відстань у площині $X0Y$ до сусіднього дрона стає меншою за встановлену критичну межу, активується алгоритм уникнення зіткнень. Щоб уникнути ситуації, коли обидва дрони одночасно змінюють висоту і знову опиняються на одному рівні, використовується правило пріоритету, згідно якого дрон лексикографічно порівнює власне локальне ім’я з ідентифікатором сусіднього. БПЛА з вищим значенням ідентифікатора, наприклад, drone1 у порівнянні з drone0, автоматично змінює цільову висоту польоту з базової на вищий ешелон. Таке правило у

поєднанні з принципами ройового методу Boids унеможливлює виникнення взаємних блокувань і дозволяє агентам безпечно розминутись у тривимірному просторі без необхідності додаткових складних перемовин чи синхронізації.

4.3.2 Модуль зв’язку та ретрансляції

Програмна логіка передачі даних і сигналів керування в багатоагентній системі реалізована через спеціалізованого агента-зв’язківця, який виконує роль інтелектуального вузла зв’язку. На відміну від звичайних статичних ретрансляторів, ці агенти використовують алгоритми ройового інтелекту для динамічного вибудовування топології мережі залежно від поточного положення дронів-пошуковців, що представлено на рисунку 4.1.

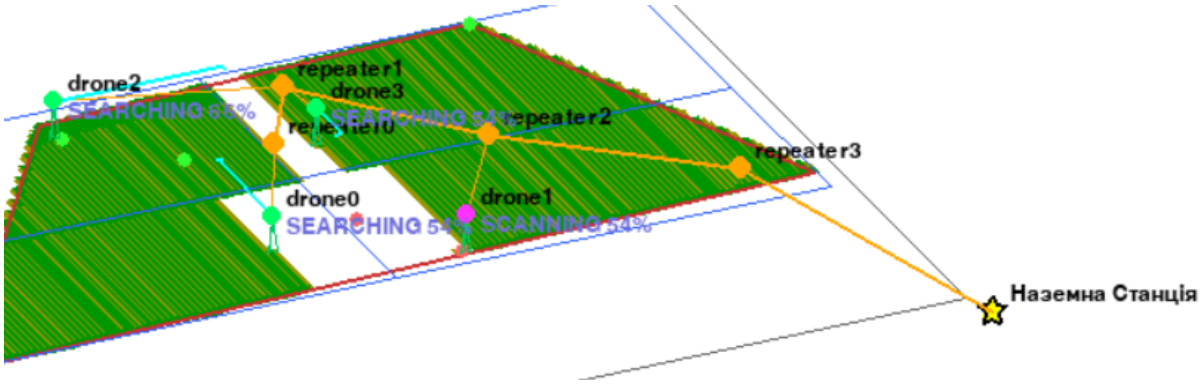


Рисунок 4.1 – Топологія локальної мережі зв’язку

Створений метод оновлення позиції дрона-зв’язківця втілює алгоритм мінімізації відхилення, що обчислює геометричний центр групи дронів-пошуковців, підключених до нього, а також враховує координати інших дронів-з’язківців у ланцюгу. Цільові координати руху зв’язківця визначаються як середнє значення між векторами групи пошуковців та підтримки зв’язку із наземною станцією керування. Такий підхід забезпечує створення адаптивного каналу з’єднання, що підлаштовується

під динамічно змінювану ситуацію. В необхідній зоні покриття забезпечується стабільний та якісний зв’язок.

4.3.3 Протокол обміну даними

Комунікація між компонентами багатоагентної системи всередині фреймворку JADE реалізована за допомогою структурованих текстових повідомлень стандарту ACL (Agent Communication Language) у поєднанні з UDP-потоками (User Datagram Protocol) для зовнішньої телеметрії.

Обмін даними починається коли людина-оператор обирає координати ділянки для обстеження та дає команду на початок роботи, що активує агента-координатора, який виступає центральною ланкою між людиною-оператором та зграєю дронів. Після отримання команди START через зовнішній UDP-порт, агент-координатор надсилає параметри багатокутної ділянки у повідомленні FIELD всім агентам-дронам.

На відміну від статичного розподілу на сектори, координатор передає повний масив точок полігону, що дозволяє дронам-пошуковцям самостійно обчислювати межі обходу території. Для підтримки синхронізації кожен агент використовує повідомлення типу INFORM з даними, які містять поточні координати та стан. Це дозволяє іншим агентам оновлювати власні внутрішні таблиці станів, що є важливим для алгоритму динамічного позиціонування дронів-зв’язківців, що підтримують стабільність зв’язку.

Обмін даними також включає підсистему протоколювання для збереження та візуалізації подій, створених агентами-пошуковцями при виявленні небезпечних об’єктів у секторі обстеження. Наприкінці роботи агентів по обстеженню території, система забезпечує відображення людині-оператору мапи місцевості із нанесеними на неї мітками місць знаходження виявлених об’єктів, їхні координати та кадри з відео та тепловізійної камер дрона.

4.4 Інтерфейс користувача

Інтерфейс користувача та модуль візуалізації розроблені як самостійний програмний компонент на мові Python з використанням бібліотеки Pygame для відображення тривимірного простору симуляції багатоагентної системи. Інтерфейс дозволяє користувачу визначити ділянку для проведення пошуку, призначити кількість дронів-пошуковців, вказати місцезнаходження наземної станції, почати виконання завдання та спостерігати за його перебігом. Загальний вигляд вікна параметрів графічного інтерфейсу наведено на рисунку 4.2.

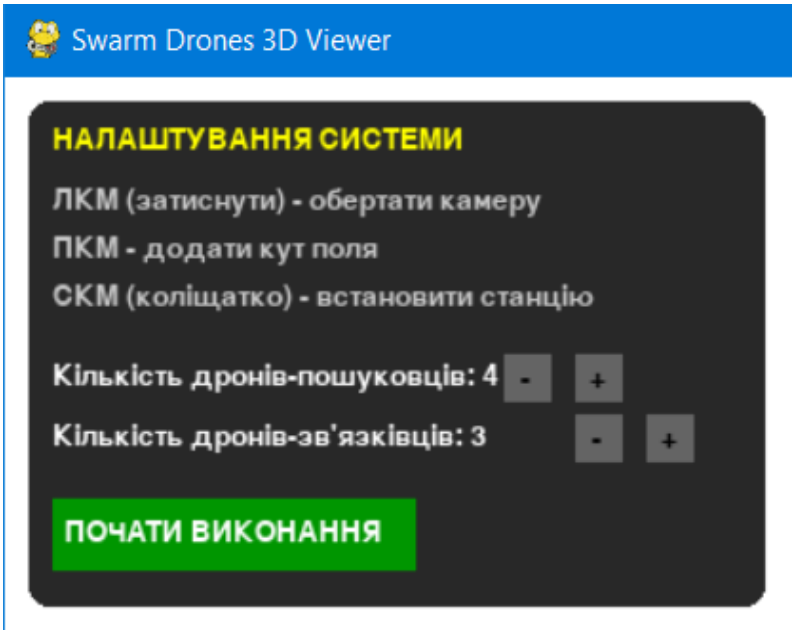


Рисунок 4.2 – Вікно параметрів інтерфейсу користувача

Після початку роботи модуль відкриває UDP-сервер для приймання телеметрії та відправляє команду ініціалізації агенту-координатору. Графічне середовище відображує поточний стан системи у реальному часі, позначаючи дронів-пошуковців та дронів-зв’язківців кольоровими сферами різного кольору та текстовими підписами, що позначають їхній поточний стан (рисунок 4.3).



Рисунок 4.3 – Відображення поточного стану в реальному часі

Зміна станів кожного з агентів відображується на екрані монітора через зміну кольорів та текстових міток.

4.5 Моделювання керування зграєю дронів при пошуку об’єктів

Процес моделювання роботи розробленої системи починається із запуску середовища виконання з ініціалізацією графічного інтерфейсу та агента-координатора. При взаємодії людини-оператора з графічним інтерфейсом Python-візуалізатора задаються межі багатокутної ділянки та вказується кількість дронів-пошуковців.

Агент-координатор отримує ці дані через UDP-порт і здійснює динамічне створення необхідної кількості екземплярів класів SearchDroneAgent та RepeaterDroneAgent безпосередньо в контейнері JADE. Формується топологія віртуального простору на основі отриманого багатокутника, показане на рисунку 4.4.

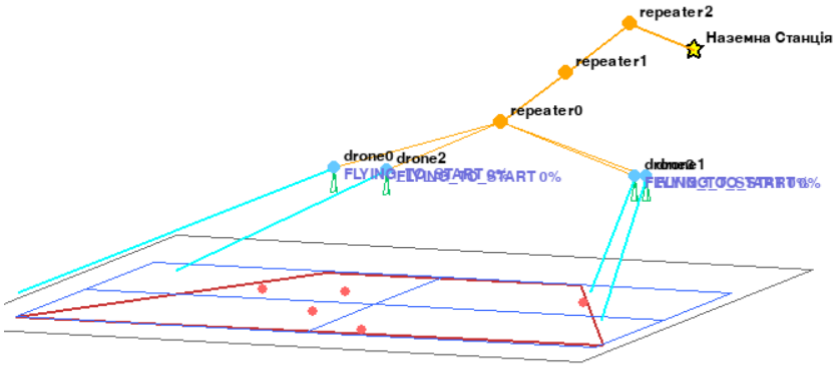


Рисунок 4.4 – Топологія віртуального простору

Генерується випадковий набір об’єктів на ділянці, які дрони-пошуковці мають знайти.

Отримавши цільові траєкторії, безпілотні летальні апарати переходять у стан активного руху за точками маршруту. Під час переміщення регулярно звіряються поточні координати з межами індивідуальних секторів сканування та передаються дані про покриття території у програму-візуалізатор. На рисунку 4.5 показана динаміка змін, що відображуються в процесі сканування дронами визначених секторів.

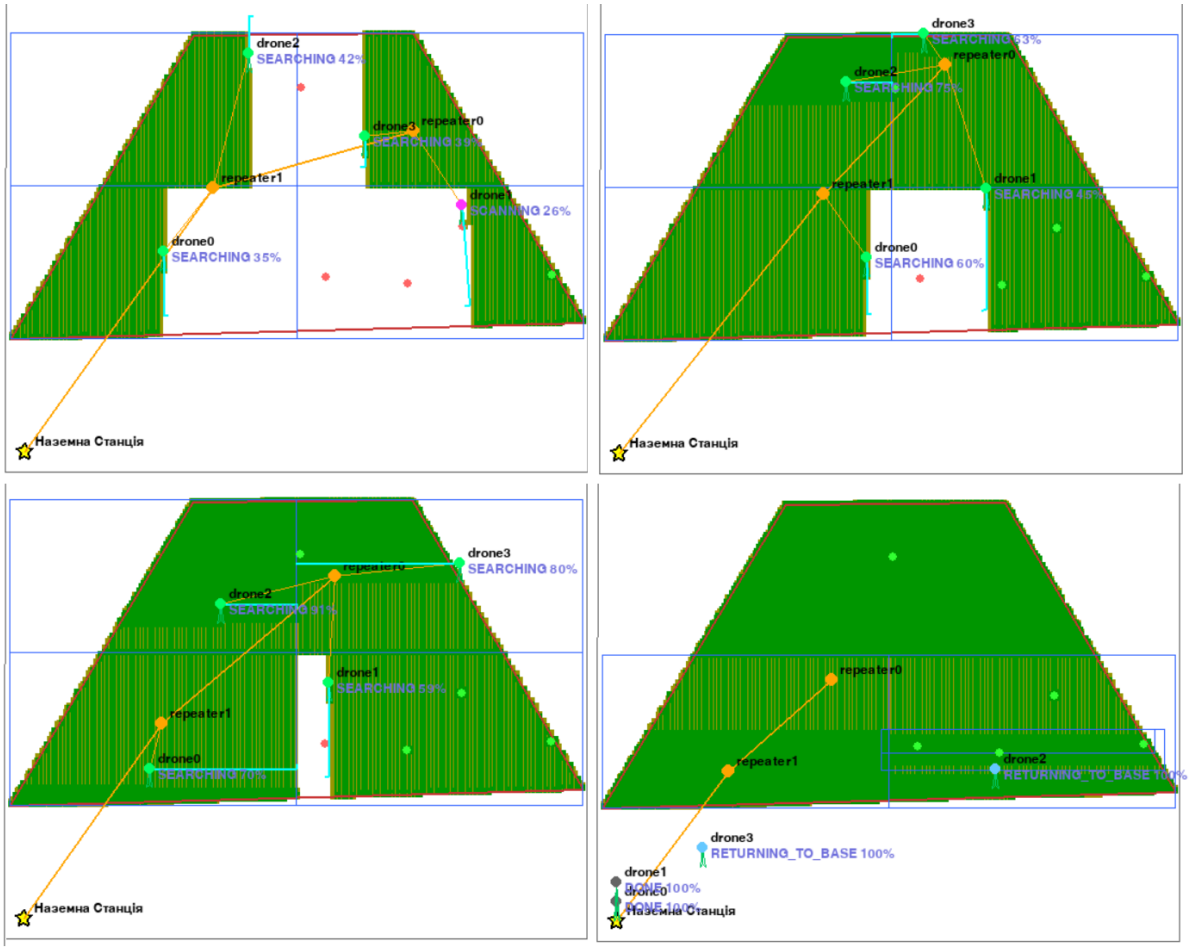


Рисунок 4.5 – Динаміка змін в процесі сканування

При виявленні об’єкта дрон облітає довкола нього, скануючи місце з різних сторін, після чого позначає об’єкт як оброблений.

Протягом усього польоту агенти підтримують актуальність розподіленої бази знань про сусідів, що дозволяє реалізовувати поведінку

уникнення зіткнень. Коли траєкторії летальних апаратів зближуються, дрон із нижчим пріоритетом автоматично змінює висоту польоту на безпечний ешелон, пропускає інший апарат і повертається до виконання робочого завдання.

За необхідності, для оптимізації розподілу часу вільний дрон-агент, що завершив сканування свого сектора, може вирушити на допомогу сусіднім агентам, що сприяє зменшенню часу обробки території та рівномірному розподілу часу та ресурсів. По закінченню сканування території зграя дронів автономно повертається до наземної станції.

Сформований звіт про проведену роботу отримується людиною-оператором, зберігається у файловій системі та візуалізується на мапі місцевості (рисунок 4.6) із мітками, що містять координати знайдених об’єктів та їх зображення з камер дронів-пошуковців.

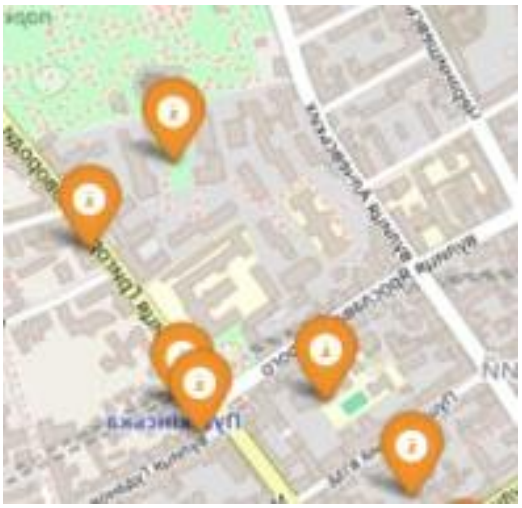


Рисунок 4.6 – Приклад мапи місцевості із нанесеними мітками виявлених об’єктів.

За необхідності людина-оператор може переглянути інформацію по кожній з нанесених на карту місцевості міток виявлених об’єктів та прив’язані до мітки дані про знайдений об’єкт (відео-та тепловізійне зображення).

4.6 Аналіз результатів моделювання

При тестування розробленої системи керування дронами-зграєю проведено комп’ютерне моделювання роботи при пошуку довільно розташованих умовно небезпечних предметів, що мають бути знайдені дронами-пошуковцями. Для цього на користувацькому інтерфейсі задана конфігурації та координати ділянки. Аналіз динаміки взаємодії в інтерактивному режимі з людиною-оператором підтверджує стабільність роботи розробленої моделі на базі JADE та ефективність організації зв’язку. Застосований алгоритм просторового розведення по висотах успішно запобігав конфліктним ситуаціям під час руху дронів.

Створена гібридна система показала високу адаптивність до складної геометрії обраної оператором ділянки, забезпечуючи її повне сканування дронами-пошуковцями, а застосовані ройові алгоритми у роботі дронів-зв’язківців допомагали ефективно підтримувати стабільний зв’язок із дронами-пошуковцями при постійному русі.

ВИСНОВКИ

У кваліфікаційній роботі досліджувались інтелектуальні агенти, як складні когнітивні системи, та методи ройового інтелекту, що активно застосовуються при керуванні групами автономних агентів. Їх використання дозволяє забезпечити масштабованість систем, високу стійкість до відмов окремих елементів та адаптивність до змін середовища. Завдяки децентралізації такі системи добре підходять для керування роями безпілотних літальних апаратів.

Розглянуті сучасні й перспективні напрямки розвитку інтелектуальних агентів. На основі проведеного огляду стану проблеми визначена мета та завдання дослідження при застосуванні методів ройового інтелекту при керуванні дронами-зграєю. Визначена доцільність розробки гібридної розподіленої архітектури на основі мультиагентної системи з ройовим інтелектом для обробки інформації в on-line режимі, адже методи ройового інтелекту в наш час є перспективним напрямом досліджень у галузі штучного інтелекту. Ці методи використовуються для задач колективної навігації, розподілу цілей між агентами, пошуку об’єктів у складному середовищі, побудови оптимальних маршрутів та підтримання формації групи. Використання подібних підходів дозволяє значно підвищити ефективність роботи систем, зокрема при керуванні зграєю БПЛА в реальних умовах.

Ефективність розробки системи на основі штучного інтелекту значною мірою залежить від вибору архітектури агентів та складності завдання, що вирішується. У ході виконання роботи був проведений аналіз області практичного застосування, оглянуто існуючі методи вирішення відповідних завдань з керування. Визначені обмеження та недоліки ручного керування та перспективність використання багатоагентного метода для практичних застосувань.

Кваліфікаційна робота є продовженням дослідження проблеми гуманітарного розмінування, розпочатих в бакалаврській роботі. З огляду на розвиток технологій розглянуто створення спеціалізованої інтелектуальної багатоагентної системи для заміни небезпечного й затратного процесу пошуку небезпечних об’єктів засобами на основі штучного інтелекту з фіксацією й протоколюванням знайдених небезпечних предметів.

Для досягнення поставленої мети досліджень було вирішено ряд завдань. Побудована система керування зграєю за ієрархічною розподіленою архітектурою. Розроблений гібридний алгоритм керування дронами-зграєю для практичного застосування з пошуку об’єктів. Застосовані спеціалізовані агенти-пошуковці, що співпрацюють для виконання спільного завдання. Забезпечується висування зграї на визначені позиції, пошук та виявлення об’єктів, передача даних та їх зберігання.

Для забезпеченням стабільності зв’язку, необхідного для безперервного й стабільного функціонування системи, створена зграя агентів-зв’язківців, що оптимізуються відповідно до необхідних динамічно змінюваних умов в реальному часі.

Координація зграї та оптимізація розподілу завдань виконується агентом-координатором. Керування рухом реалізоване на основі моделі Boids, що є базовою для безпечності руху зграї із забезпеченням локальної взаємодії, тоді як ройові алгоритми забезпечують оптимізацію рішень щодо забезпечення зв’язку. При цьому виконується збір даних за допомогою сенсорів, їх багаторівнева обробка з використанням агентного ШІ на основі навченої нейронної мережі CNN з автоматичним розпізнаванням об’єктів, та збереження результату у зручній для користувача формі.

Для комп’ютерного моделювання проведено вибір програмних та апаратних засобів реалізації створеної багатоагентної системи. Розроблено

програму, що реалізує систему пошуку об’єктів, із застосуванням мов програмування Java та Python. Для підтримки взаємодії з людиною-оператором розроблений графічний інтерфейс, що використовується для створення завдання для зграї з визначенням їх кількості та параметрів зони обстеження. Під час роботи оператор може спостерігати за роботою всіх дронів та прогресом сканування по окремим визначеним секторам місцевості. Сформовані за результатами обстеження дані, що надходять у реальному часі, протоколюються зі збереженням координат місцевості, ідентифікованих об’єктів та відображуються в графічному інтерфейсі оператора. Тож, розроблена система забезпечує автономне виконання завдання від отримання координат сектора до протоколювання й відображення результатів виконаного завдання, гарантуючи безперебійність роботи та високу надійність.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Lazariev V.O., Lazarieva O.O. Application of artificial intelligence in unmanned aircrafts for humanitarian needs: матеріали XI Всеукраїнської наук.-практ. конф. з автоматич. управління, Збірник наукових праць / Під редакцією Г.В. Рудакової та ін. Херсон-Хмельницький: Видавництво ФОП Вишемирський В.С., 2024. С. 69.

2. Приставка П.О., Сорокопуд В.І., Рябий М.О., Смірнов О.А. Рекомендації по архітектурі БПЛА для використання на борту відео-аналітики та навігації. *Актуальні питання забезпечення кібербезпеки та захисту інформації*: матеріали VII Міжнар. наук.-практ. конф., м. Київ, 24–27 лют. 2021 р. / Європейський університет. Київ, 2021. С. 75–77.

3. Посвістак В., Мірошніченко Д. Архітектура системи автономного керування для fpv-дронів. *Вісник Хмельницького національного університету*. 2024. Т. 337, №3(2). С. 223–230. URL: https://doi.org/10.31891/2307-5732-2024-337-3-33 (дата звернення: 15.05.2026).

4. Архітектура систем управління БПЛА: компоненти, зв’язок і принципи роботи. *BlueBird Tech.* 2026. URL: https://www.blue-bird.tech/news/arhitektura-system-upravlinnya-bpla/ (дата звернення: 15.05.2026).

5. Multi-Agent reinforcement learning aided intelligent UAV swarm for target tracking / Z. Xia et al. *IEEE transactions on vehicular technology*. 2022. Vol. 71, no. 1. pp. 931–945. URL: https://doi.org/10.1109/tvt.2021.3129504 (date of access: 15.05.2026).

6. Xia B., Mantegh I., Xie W. Decentralized UAV swarm control: a multi-layered architecture for integrated flight mode management and dynamic target interception. *Drones*. 2024. Vol. 8, no. 8, P. 350. URL: https://doi.org/10.3390/drones8080350 (date of access: 15.05.2026).

7. Multi-Target pursuit by a decentralized heterogeneous UAV swarm using deep multi-agent reinforcement learning / M. Kouzeghar et al. 2023 *IEEE international conference on robotics and automation*, London, United Kingdom, 29 May – 2 June 2023: URL: https://doi.org/10.1109/icra48891.2023.10160919 (date of access: 15.05.2026).

8. Zhou W., Li J., Zhang Q. Joint communication and action learning in multi-target tracking of UAV swarms with deep reinforcement learning. *Drones*. 2022. Vol. 6, no. 11. P. 339. URL: https://doi.org/10.3390/drones6110339 (date of access: 15.05.2026).

9. Vlassis N. A concise introduction to multiagent systems and distributed artificial intelligence. Cham: Springer International Publishing, 2007. URL https://doi.org/10.1007/978-3-031-01543-4 (date of access: 15.05.2026).

10. Russell S., Norvig P. Artificial intelligence: a modern approach, 4th US ed. URL: http://aima.cs.berkeley.edu/ (date of access: 15.05.2026).

11. Krishnan N. AI agents: evolution, architecture, and real-world applications. *arXiv*. 2025. URL: https://doi.org/10.48550/arXiv.2503.12687 (date of access: 15.05.2026).

12. Multi-Agent collaboration mechanisms: a survey of LLMs / K.-T. Tran et al. *arXiv*. 2024. URL: https://doi.org/10.48550/arXiv.2501.06322 (date of access: 15.05.2026).

13. Laird J.E., Lebiere C., Rosenbloom P.S. A standard model of the mind: toward a common computational framework across artificial intelligence, cognitive science, neuroscience, and robotics. *AI magazine*. 2017. Vol. 38, no. 4. pp. 13–26. URL: https://doi.org/10.1609/aimag.v38i4.2744 (date of access: 16.05.2026).

14. Gomez V. What is Agentic AI? A Guide. *Capicua*. 2026. URL: https://www.capicua.com/blog/what-is-agentic-ai (date of access: 16.05.2026).

15. Методичні вказівки до виконання практичних робіт з дисципліни «Мультиагентні системи» для студ. спец. 7.080402 «Інформаційні

технології проектування» / С.О. Субботін, О.О. Олійник, А.О. Олійник. – Запоріжжя: ЗНТУ, 2008. 18 с.

16. Advances and challenges in foundation agents: from brain-inspired intelligence to evolutionary, collaborative, and safe systems. 2025. *arXiv* URL: https://doi.org/10.48550/arXiv.2504.01990. (date of access: 28.03.2026).

17. Tree of thoughts: deliberate problem solving with large language models / S. Yao et al. *arXiv*. 2023. URL: https://doi.org/10.48550/arXiv.2305.10601 (date of access: 16.05.2026).

18. The rise and potential of large language model based agents: a survey / Z. Xi et al. *Science China information sciences*. 2025. Vol. 68, no. 2. P. 121101. URL: https://doi.org/10.1007/s11432-024-4222-0 (date of access: 16.05.2026).

19. ReAct: synergizing reasoning and acting in language models / S. Yao et al. *GitHub*. URL: https://react-lm.github.io/ (date of access: 15.05.2026).

20. A distributed architecture for control of autonomous mobile robots / P. Baroni et al. URL: https://pietro-baroni.unibs.it/PDFpapers/CI-4.pdf (date of access: 15.05.2026).

21. Лазарєв В. Огляд методів ройового інтелекту для керування групою автономних об’єктів. *Комп’ютерні інтелектуальні системи та мережі*: матеріали XIX Всеукр. наук. практ. WEB конф. аспірантів, студентів та молодих вчених, м. Кривий Ріг, 25–27 березня 2026 р.). / Криворізький національний університет. Кривий Ріг, 2026. С. 211–214.

22. Kennedy J., Eberhart R. Particle swarm optimization // *Proceedings of International Conference on Neural Networks (ICNN'95)*. Vol. 4. pp. 1942–1948. URL: https://doi.org/10.1109/ICNN.1995.488968 (date of access: 16.05.2026)

23. Dorigo M., Gambardella L.M. Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE transactions on evolutionary computation*. 1997. Vol. 1, no. 1. pp. 53–66. URL: https://doi.org/10.1109/4235.585892 (date of access: 16.05.2026).

24. Karaboga D. An idea based on honey bee swarm for numerical optimization. *Erciyes University: Technical Report TR06*. 2005. URL: https://abc.erciyes.edu.tr/pub/tr06_2005.pdf (date of access: 16.05.2026).

25. Reynolds C.W. Flocks, herds and schools: a distributed behavioral model. *ACM SIGGRAPH computer graphics*. 1987. Vol. 21, no. 4. pp. 25–34. URL: https://doi.org/10.1145/37402.37406 (date of access: 16.05.2026).

26. Brodie J. Autonomous agents in autonomous worlds. *Medium*. 2024. URL: https://medium.com/id-theory/autonomous-agents-in-autonomous-worlds-2f1853f8a1d7 (date of access: 16.05.2026).

27. Hu J. A survey on reinforcement learning-based multi-agent path planning. *ITM web of conferences*. 2025. Vol. 78. P. 01015. URL: https://doi.org/10.1051/itmconf/20257801015 (date of access: 16.05.2026).

28. Fast multi-uav path planning for optimal area coverage in aerial sensing applications / M.A. Luna et al. *Sensors*. 2022. Vol. 22, no. 6. P. 2297. URL: https://doi.org/10.3390/s22062297 (date of access: 16.05.2026).

29. Роботько С., Топалов А. Багаторівнева архітектура системи автоматичного керування БПЛА для здійснення пошукових місій за відеоаналізом та металодетекцією. *Оптико-електронні інформаційно-енергетичні технології*. 2026. вип. 50. № 2. С. 114–124. URL: https://doi.org/10.31649/1681-7893-2025-50-2-114-124 (дата звернення: 15.05.2026)

30. How do classification models empower agentic AI's decision-making intelligence? *Monetizely*. 2025. URL: https://www.getmonetizely.com/articles/how-do-classification-models-empower-agentic-ais-decision-making-intelligence (date of access: 15.05.2026).

31. Gradient-based learning applied to document recognition / Y. Lecun et al. *Proceedings of the IEEE*. 1998. Vol. 86, no. 11. pp. 2278–2324. URL: https://doi.org/10.1109/5.726791 (date of access: 15.05.2026).

32. Gurucharan M.K. Basic CNN architecture: explaining 5 layers of convolutional neural network. *UpGrad*. 2026. URL: https://www.upgrad.com/blog/basic-cnn-architecture/ (date of access: 15.05.2026).

33. Deep Deep learning for computer vision. *GeeksforGeeks*. 2026. URL: https://www.geeksforgeeks.org/computer-vision/deep-learning-for-computer-vision/ (date of access: 15.05.2026).

34. YOLO-world model. *Ultralytics*. URL: https://docs.ultralytics.com/models/yolo-world (date of access: 15.05.2026).

35. Explore ultralytics YOLOv8. *Ultralytics*. URL: https://docs.ultralytics.com/models/yolov8 (date of access: 15.05.2026).

36. Point in polygon. *Wikipedia*. URL: https://en.wikipedia.org/wiki/Point_in_polygon (date of access: 15.05.2026).

37. Sunday D. Inclusion of a point in a polygon. URL: https://web.archive.org/web/20130126163405/http:/geomalgorithms.com/a03-_inclusion.html (date of access: 15.05.2026).

38. Datagrams in Java. *Geeksforgeeks*. 2025. URL: https://www.geeksforgeeks.org/java/datagrams-in-java/ (date of access: 15.05.2026).

39. Datagram in Python. *Geeksforgeeks*. 2019. URL: https://www.geeksforgeeks.org/python/datagram-in-python/ (date of access: 15.05.2026).