МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
«КИЇВСЬКИЙ АВІАЦІЙНИЙ ІНСТИТУТ»

Кваліфікаційна праця
на правах рукопису

МИРОШНИЧЕНКО ІГНАТ ВАСИЛЬОВИЧ

УДК 621.391.83(045)

ДИСЕРТАЦІЯ
МОДЕЛЬ НЕЙРОМЕРЕЖЕВОГО УПРАВЛІННЯ ГРУПОЮ БПЛА

122 «Комп’ютерні науки»
12 «Інформаційні технології»
Подається на здобуття ступеня доктора філософії

Дисертація містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело. _____І.В. Мирошниченко

Науковий керівник:
Кучеров Дмитро Павлович
доктор технічних наук, професор

АНОТАЦІЯ

Мирошниченко І.В. Модель нейромережевого управління групою БПЛА. – кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття ступеня доктора філософії в галузі знань 12 «Інформаційні технології», за спеціальністю 122 «Комп'ютерні науки». – Національний університет «Київський авіаційний інститут», м. Київ, 2026.

Дисертаційна робота присвячена розробці та верифікації децентралізованої нейромережевої моделі управління групою безпілотних літальних апаратів (БПЛА), що водночас забезпечує точність утримання формації, гарантії безпеки між агентами рою та масштабованість без перенавчання при зміні кількості апаратів.

Актуальність дослідження зумовлена зростаючою потребою у координованих групах БПЛА (5-20 апаратів) у цивільних і оборонних застосуваннях, зокрема в умовах збройного конфлікту в Україні. Наявні підходи правила Рейнольдса, метод потенційних полів (APF), мульти-агентного навчання з підкріпленням для корпоративного середовища (MADDPG), глибокого навчання з підкріпленням (QMIX) вирішують задачу групового управління лише частково: або не забезпечують точності формації $\Delta f \leq 0{,}20$ м, або не масштабуються без повторного навчання при зміні N, або не містять явних механізмів гарантування безпечних відстаней між апаратами. Поєднання графової нейронної мережі з механізмом уваги (Graph Attention Network, GAT) та алгоритму мультиагентного навчання з підкріпленням MAPPO за парадигмою централізованого навчання/децентралізованого виконання (CTDE) визначено як теоретично обґрунтований підхід до вирішення зазначеної тріади вимог.

Об'єктом дослідження є процес децентралізованого управління групою автономних БПЛА в умовах обмеженого зв'язку, динамічної топології взаємодій та вимог безпеки між агентами рою.

Предметом дослідження є графові нейронні мережі (GNN, GAT) та методи мультиагентного навчання з підкріпленням (MARL, MAPPO) для координації рою БПЛА.

У першому розділі виконано систематичний огляд та порівняльний аналіз публікацій щодо методів координації груп БПЛА: правила Рейнольдса, метод потенційних полів, MADDPG, QMIX, MAPPO, GNN-підходи. Встановлено, що жоден з існуючих методів не забезпечує одночасно $\Delta f \leq 0{,}20$ м, $CR_{inter} < 2\%$ та масштабованість від $N = 3$ до $N = 10$ без перенавчання. Сформульовано наукову задачу у форматі кооперативного децентралізованого частково спостережуваного процесу прийняття рішень Маркова (Dec-POMDP) та визначено технічні вимоги до системи управління групою БПЛА.

У другому розділі розроблено формальну математичну модель N-агентної системи БПЛА як динамічного графа $G(t) = (V, E(t))$, де ребра визначаються радіусом комунікації R_{comm}. Вектор стану агента $s_i \in R^{12}$ включає положення, лінійну швидкість, кути Ейлера та кутові швидкості. Протокол повідомлень $m_{ji} \in R^7$ містить відносне положення сусіда, його швидкість та норму відносного зсуву у формації. Введено метрику масштабованості $EI = SR/N$, де SR – частота успішних місій, N – кількість БПЛА, та функцію безпеки на основі умови мінімальної відстані d_{safe} між агентами.

У третьому розділі запропоновано GNN-архітектуру моделі централізованого навчання та децентралізованого виконання (CTDE), що складається з двох компонентів: Актора уваги на графі (GATv2Conv, 2 шари, 4 голови уваги, ~ 29 800 параметрів) для децентралізованого виконання на борту кожного БПЛА з частотою 20 Гц та Критика уваги на графі з глобальним об'єднанням для централізованого навчання. Обчислювальна складність Actor становить $O(N*k)$ замість $O(N^2)$ у конкатенаційних підходах. Функція винагороди MAPPO включає чотири компоненти: штрафи за похибку формації, за порушення безпечної відстані, за втрату зв'язності графа та доданок ефективності. Реалізовано двоступеневий механізм гарантування безпеки: репульсивний потенціал у вигляді функції винагороди ($\beta/\alpha = 8$) та явний шар забезпечення безпечних відстаней на основі квадратичного програмування (QP Safety Layer) з оператором розщіплення вирішувача на основі квадратичного програмування (OSQP-вирішувач) на рівні виконання (витратність ~0,5 мс).

У четвертому розділі наведено результати симуляційної верифікації у середовищі Gazebo 11 (ArduPilot SITL, 5 сценаріїв, $N \in \{3, 5, 10\}$, по 30 запусків). GNN+MAPPO досягає: середньої похибки формації $\Delta f = 0{,}14$ м (на 50% кращої за MADDPG та на 73% -- за правила Рейнольдса); частоти зіткнень між агентами $CR_{inter} = 1{,}2\%$ (< 2%); частоти успішних місій $SR = 93\%$; часу збору формації $T_{form} = 5{,}5$ с.

Усі п'ять цільових вимог виконано. Статистична значущість переваги підтверджена за критерієм Вілкоксона ($p < 0{,}001$). Масштабованість верифіковано: при тестуванні на $N = 10$ після навчання на $N = 5$ деградація SR становить лише 5,4 п.п. (93% → 88%), що у 8 разів менше ніж у MADDPG. Actor-мережа (INT8, ~30 КБ) виконується за 10,7 ± 1,2 мс на ARM Raspberry Pi 4, що у 77 разів швидше за централізований (MPC).

Наукова новизна одержаних результатів полягає у такому:

- вперше розроблено метод децентралізованого просторового управління роєм БПЛА, який, за рахунок застосування графової нейронної мережі з механізмом уваги на основі парадигми централізованого навчання та децентралізованого виконання з інваріантним до перестановок агрегуванням повідомлень між агентами, що замінює конкатенацію глобальних станів та забезпечує обчислювальну лінійну складність актора замість квадратичної, що дозволяє досягти масштабованості рою БПЛА без перенавчання за одночасного підвищення точності утримання формації і часу бортового виконання на ARM-процесорі.;
- удосконалено метод мультиагентного навчання з підкріпленням, який за рахунок введення «Актора» та «Критика» уваги на графі з глобальним об'єднанням замість конкатенації станів агентів, що дозволяє знизити обчислювальну складність алгоритму з квадратичної до лінійної та забезпечити стійкість навчання при зміні кількості агентів у рої БПЛА;
- удосконалено підхід до забезпечення безпечної взаємодії агентів рою БПЛА за рахунок впровадження дворівневого механізму гарантування безпеки, який поєднує формування функції винагороди з урахуванням міжагентних

відстаней та корекцію керуючих дій на основі квадратичного програмування, що дозволяє забезпечити дотримання безпечних дистанцій між агентами та знизити ризик міжагентних зіткнень під час виконання групових місій;

- набуло подальшого розвитку метрики оцінювання масштабованості алгоритмів координації груп БПЛА, які, за рахунок уніфікованого формування на основі частоти успішних місій та набору тестових сценаріїв з нормуванням (центруванням) до еталону, дозволяють об'єктивно порівнювати робастність різних алгоритмів координації за змінної кількості агентів.

Практичне значення:

- розроблено програмний прототип моделі ройового управління (~4 000 рядків Python, PyTorch 2.0, PyG 2.3, ROS Noetic) з програмним пакетом підтримки розрахунків для середовищ сумісних з мовою Python та MAVLink-інтерфейсом до програмної системи ArduPilot, який придатний до застосування у системах координованого картографування, що дозволяє підвищити покриття у 8-10 разів краще за одиночний апарат при виконанні пошуково-рятувальних операцій та оборонних БПЛА-місіях.

За темою дисертаційного дослідження опубліковано 13 наукових праць, у тому числі: 2 публікацій у наукових виданнях, що індексуються у міжнародній наукометричній базі Scopus; 4 статей у фахових виданнях України категорії Б; 1 стаття у міжнародному журналі; 6 тез доповідей на міжнародних та всеукраїнських конференціях.

Ключові слова: безпілотне літальний апарат, рій, безпілотні літальні апарати, БПЛА, децентралізоване управління, нейронна мережа, навчання з підкріпленням, навчання з вчителем, аналіз ефективності, машинне навчання, навігаційна система, прийняття рішень, штучний інтелект, надійність, алгоритми рою.

ABSTRACT

Myroshnychenko I.V. Neural Network Model for UAV Swarm Control. Qualifying scientific work as a manuscript.

Dissertation for the degree of Doctor of Philosophy in the field of knowledge 12 «Information Technologies», specialty 122 «Computer Science». – National University "Kyiv Aviation Institute", Kyiv, 2026.

The dissertation is devoted to the development and verification of a decentralized neural network control model for a group of unmanned aerial vehicles (UAVs) that simultaneously ensures formation-keeping accuracy, inter-agent safety guarantees within the swarm, and scalability without retraining when the number of vehicles changes.

The relevance of the research is driven by the growing need for coordinated UAV groups (5–20 vehicles) in civilian and defense applications, particularly in the context of the armed conflict in Ukraine. Existing approaches – Reynolds' rules, the Artificial Potential Field (APF) method, Multi-Agent Deep Deterministic Policy Gradient (MADDPG), and value decomposition via QMIX – address the swarm control problem only partially: they either fail to achieve formation accuracy of $\Delta f \leq 0.20$ m, cannot scale without retraining when N changes, or lack explicit mechanisms for guaranteeing safe inter-agent distances. The combination of a Graph Attention Network (GAT) with a Multi-Agent Proximal Policy Optimization (MAPPO) algorithm under the Centralized Training with Decentralized Execution (CTDE) paradigm has been identified as a theoretically grounded approach to addressing this triad of requirements.

The object of the research is the process of decentralized control of a group of autonomous UAVs under conditions of limited communication, dynamic interaction topology, and inter-agent safety requirements.

The subject of the research is graph neural networks (GNN, GAT) and multi-agent reinforcement learning methods (MARL, MAPPO) for swarm coordination.

In the first chapter, a systematic review and comparative analysis of publications on UAV group coordination methods is carried out: Reynolds' rules, the potential field

method, MADDPG, QMIX, MAPPO, and GNN-based approaches. It is established that none of the existing methods simultaneously achieves $\Delta f \leq 0.20$ m, $CR_{inter} < 2\%$, and scalability from $N = 3$ to $N = 10$ without retraining. The scientific problem is formulated as a cooperative decentralized partially observable Markov decision process (Dec-POMDP), and the technical requirements for the UAV group control system are defined.

In the second chapter, a formal mathematical model of an N-agent UAV system is developed as a dynamic graph $G(t) = (V, E(t))$, where edges are defined by a communication radius R_comm. The agent state vector $s_i \in R^{12}$ includes position, linear velocity, Euler angles, and angular velocities. The message protocol $m_{ji} \in R^7$ encodes the neighbor's relative position, velocity, and the norm of the relative formation offset. A scalability metric $EI = SR/N$ is introduced, where SR is the mission success rate and N is the number of UAVs, along with a safety function based on the minimum inter-agent distance condition d_{safe}.

In the third chapter, a GNN-based CTDE architecture is proposed, comprising two components: a Graph Attention Actor (GATv2Conv, 2 layers, 4 attention heads, ~29,800 parameters) for decentralized onboard execution on each UAV at 20 Hz, and a Graph Attention Critic with global pooling for centralized training. The computational complexity of the Actor is $O(N*k)$ rather than $O(N^2)$ as in concatenation-based approaches. The MAPPO reward function incorporates four components: penalties for formation error, violations of safe distance, loss of graph connectivity, and an efficiency bonus. A two-level safety assurance mechanism is implemented: a repulsive potential in the reward function ($\beta/\alpha = 8$) and an explicit Quadratic Programming Safety Layer (QP Safety Layer) with an OSQP solver at the execution level (overhead ~0.5 ms).

In the fourth chapter, the results of simulation-based verification in the Gazebo 11 environment (ArduPilot SITL, 5 scenarios, $N \in \{3, 5, 10\}$, 30 runs each) are presented. GNN+MAPPO achieves: a mean formation error of $\Delta f = 0.14$ m (50% better than MADDPG and 73% better than Reynolds' rules); an inter-agent collision rate of

CR_{inter} = 1.2% (< 2%); a mission success rate of SR = 93%; and a formation assembly time of T_{form} = 5.5 s.

All five target requirements are met. Statistical significance of the performance advantage is confirmed by the Wilcoxon signed-rank test ($p < 0.001$). Scalability is verified: when tested on N = 10 after training on N = 5, the SR degradation is only 5.4 percentage points (93% → 88%), which is 8 times less than that of MADDPG. The Actor network (INT8, ~30 KB) executes in 10.7 ± 1.2 ms on an ARM Raspberry Pi 4, which is 77 times faster than centralized Model Predictive Control (MPC).

Scientific novelty of the obtained results consists in the following:

- for the first time, a method of decentralized spatial control of a UAV swarm has been developed, which, by using a graph neural network with an attention mechanism based on the paradigm of centralized learning and decentralized execution with permutation-invariant aggregation of messages between agents, replaces the concatenation of global states and provides a linear computational complexity of the actor instead of a quadratic one, which allows to achieve scalability of the UAV swarm without retraining while simultaneously increasing the accuracy of formation maintenance and on-board execution time on an ARM processor..
- a method of multi-agent reinforcement learning has been improved, which, by introducing the "Actor" and "Critic" attention on a graph with a global union instead of concatenation of agent states, allows to reduce the computational complexity of the algorithm from quadratic to linear and ensure the stability of learning when changing the number of agents in the UAV swarm.
- the approach to ensuring safe interaction of UAV swarm agents has been improved by implementing a two-level security guarantee mechanism, which combines the formation of a reward function taking into account inter-agent distances and the correction of control actions based on quadratic programming, which allows ensuring compliance with safe distances between agents and reducing the risk of inter-agent collisions during group missions.
- the metrics for assessing the scalability of UAV group coordination algorithms have been further developed, which, due to the unified formation based on

the frequency of successful missions and a set of test scenarios with normalization (centering) to the standard, allow objectively comparing the robustness of different coordination algorithms with a variable number of agents.

Practical significance:

- A software prototype of a swarm control model has been developed (~4,000 lines of Python, PyTorch 2.0, PyG 2.3, ROS Noetic) with a computational support package for Python-compatible environments and a MAVLink interface to the ArduPilot software system, suitable for use in coordinated mapping systems, enabling 8–10× better coverage than a single vehicle in search-and-rescue operations and defensive UAV missions.

On the topic of the dissertation research, 13 scientific works have been published, including: 2 publications in journals indexed in the international scientometric database Scopus; 4 articles in professional Ukrainian journals of Category B; 1 article in an international journal; 6 conference abstracts presented at international and all-Ukrainian conferences.

Keywords unmanned aerial vehicle, swarm, unmanned aerial vehicles, UAV, decentralized control, neural network, reinforcement learning, supervised learning, efficiency analysis, machine learning, navigation system, decision-making, artificial intelligence, reliability, swarm algorithm.

СПИСОК ПРАЦЬ ЗДОБУВАЧА

Статті у наукових фахових виданнях:

1. Кучеров Д. П., Перепеліцин С.О., Пошивайло О.М., Мирошниченко І.В. Налаштування ПІД-регулятора генетичним алгоритмом за багатокритеріальною цільовою функцією для керування нестійким об’єктом. *Проблеми інформатизації та управління,* 2023. №4(76). С.42-47. DOI: https://doi.org/10.18372/2073-4751.76.18239

Особистий внесок автора: програмна реалізація алгоритму та участь у проведенні чисельних експериментів.

Особистий внесок Кучерова Д.П.: постановка наукової задачі, формування загальної концепції дослідження.

Особистий внесок Перепеліцина С.О.: розробка математичної моделі об’єкта керування та формалізація задачі оптимізації.

Особистий внесок Пошивайла О.М.: реалізація генетичного алгоритму для налаштування параметрів ПІД-регулятора та аналіз результатів моделювання.

2. Мирошниченко І. В. Інформаційних технологій в системах управління безпілотних літальних апаратів. *Herald of Khmelnytskyi National University. Technical Sciences*. 2025. №*357*(5.1). С. 299-306. DOI: https://doi.org/10.31891/2307-5732-2025-357-38

3. Мирошниченко І.В. Інтегровані навігаційні та комунікаційні протоколи для автономних інтелектуальних систем: архітектури, моделі та виклики реалізації в динамічних середовищах. *Інформаційні технології та суспільство*. 2025. №2 (17). С. 116-122. DOI: https://doi.org/10.32689/maup.it.2025.2.17

4. Myroshychenko I. Multi-agent Control of UAVs Using Deep Reinforcement Learning. *Electronics and Control Systems*. 2025. № *2*(84). С. 15–20. DOI: https://doi.org/10.18372/1990-5548.84.20187

Стаття у виданні, яке включено до міжнародних наукометричних баз:

1. Myroshnychenko I., Kucherov D., Dolgikh, S., Kondratyuk V. Navigation and Communications Protocols for Autonomous Intelligent Mobility. *Proceedings - International Conference on Advanced Computer Information Technologies, ACIT*. 2024. DOI: https://doi.org/10.1109/acit62333.2024.10712508 **(індексується базою Scopus).**

Особистий внесок автора: розробка моделей протоколів навігації та комунікації для автономної інтелектуальної мобільності, аналіз ефективності функціонування автономної системи

Особистий внесок Kucherov D.: постановка задачі дослідження, розробка концепції протоколів автономної інтелектуальної мобільності, формалізація функціональних рівнів системи.

Особистий внесок Dolgikh S: розробка математичних моделей автономної системи та визначення вимог до її функціонування.

Особистий внесок Kondratyuk S.: дослідження комунікаційних процесів та моделей обміну інформацією між агентами.

2. Kucherov D., Dolgikh S., Myroshnychenko I., Poshyvailo O., Kravchenko O. A flight situation advisory system for uninterrupted and efficient air transportation. *IOP Conf. Ser.: Earth Environ.* 2024. DOI: https://doi.org/10.1088/1755-1315/1415/1/012035 **(індексується базою Scopus).**

Особистий внесок автора: формалізація польотних ситуацій, розробка моделей оцінювання пріоритетів на основі даних сенсорів, участь у побудові математичного апарату бортової рекомендаційної системи та аналіз результатів дослідження .

Особистий внесок Kucherov D.: постановка наукової проблеми, формування загальної концепції системи та визначення критеріїв ефективності.

Особистий внесок Dolgikh S.: розробка архітектури системи та підходів до прийняття рішень в умовах невизначеності.

Особистий внесок Poshyvailo O.: програмна реалізація окремих компонентів системи та участь у проведенні експериментальних досліджень.

Особистий внесок Kravchenko O.: аналіз показників безпеки, надійності та безперервності функціонування авіаційної системи.

Наукові праці, які додатково відображають наукові результати дисертації:

1. Kucherov D., Poshyvailo O., Myroshnychenko I. Simulation of PID-Regulator Tuning Using Genetic Algorithm on Multi-Criteria Objective Function for Controlling Unstable Objects. *CEUR Workshop Proceedings*. 2023. Vol. 3887. P. 108-118.

Особистий внесок автора: участь у розробці підходу з використанням генетичного алгоритму, аналіз ефективності отриманих результатів моделювання.

Особистий внесок Kucherov D.: постановка наукової задачі, визначення загальної концепції методу налаштування ПІД-регулятора та вибір критеріїв якості, формалізація багатокритеріальної цільової функції налаштування ПІД-регулятора

Особистий внесок Poshyvailo O.: програмна реалізація генетичного алгоритму, проведення чисельних експериментів та візуалізація результатів.

2. Myroshnychenko I., Kucherov D. Approach for controlling a group of unmanned aerial vehicles using neural network. *«AVIA-2023» міжнародна науково-технічна конференція*: тези доп., 18-20 квіт, 2023. Київ, Україна. С. 10.1-10.5. URL: https://conference.nau.edu.ua/index.php/AVIA/AVIA2023/paper/view/9268/7607

Особистий внесок автора: розробка наукового підходу до управління групою безпілотних літальних апаратів на основі нейромережевих методів, формалізація задачі групового управління, побудова архітектури нейронної мережі, обґрунтування вибору методу навчання та аналіз отриманих результатів.

Особистий внесок Kucherov D.: участь у постановці задачі, наукове консультування та узагальнення результатів дослідження.

3. Kucherov D., Dolgikh S., Myroshnychenko I., Poshyvailo O. Protocol and Logical Models for Robustness and Survivability in Autonomous UAV Systems. 2023 13th International Conference on Dependable Systems, Services and Technologies (DESSERT): conf. paper, 13-15 oct. 2023. Athens, Greece, 2023, P. 1-7, DOI: https://doi.org/10.1109/dessert61349.2023.10416491

Особистий внесок автора: участь у формалізації поняття робастності автономних систем, розробці матричної моделі станів та аналізі результатів дослідження.

Особистий внесок Kucherov D.: постановка наукової задачі, формування загальної концепції дослідження.

Особистий внесок Dolgikh S.: дослідження показників надійності та відмовостійкості систем.

Особистий внесок Poshyvailo O.: програмна реалізація моделей та проведення експериментів.

4. Kucherov D., Poshyvailo O., Myroshnychenko I. Dynamics of UAV Discrete-Continuous Terminal Control System, *2023 IEEE 7th International Conference on Methods and Systems of Navigation and Motion Control (MSNMC)*: conf. paper, 24-27 oct, 2023 Kyiv, Ukraine. 2023. P. 66-69, DOI: https://doi.org/10.1109/msnmc61017.2023.10329065

Особистий внесок автора: участь у розробці моделі дискретно-неперервної системи керування БПЛА, аналіз умов формування перемикальної функції та інтерпретація результатів моделювання.

Особистий внесок Kucherov D.: постановка задачі синтезу системи керування та формування загальної концепції дослідження.

Особистий внесок Poshyvailo O.: розробка алгоритму керування та реалізація моделювання системи.

5. Kucherov D., Shmelova T., Poshyvailo O., Tkachenko V., Myroshnychenko I., Ogirko I. Mathematical Model of Damping of UAV Oscillations

in the Cargo Delivery Problem, *2023 IEEE 4th KhPI Week on Advanced Technology (KhPIWeek):* conf. paper,, 02-06 Oct. 2023. Kharkiv, Ukraine, 2023, P. 1-6, DOI: https://doi.org/10.1109/khpiweek61412.2023.10312855

Особистий внесок автора: реалізація програмного забезпечення для моделювання, аналіз та інтерпретація результатів симуляцій.

Особистий внесок Kucherov D.: постановка наукової проблеми, розробка загальної концепції моделювання коливань вантажу, формулювання задачі керування.

Особистий внесок Shmelova T.: математичне моделювання динаміки БПЛА, забезпечення аналітичного апарату для контролерів.

Особистий внесок Poshyvailo O.: розробка та оптимізація PID-контролера для невеликих кутових відхилень, симуляційні експерименти.

Особистий внесок Tkachenko V.: розробка контролера для великих кутових відхилень, участь у координації алгоритмів керування.

Особистий внесок Ogirko I.: реалізація програмного забезпечення для моделювання, підготовка та обробка даних для симуляцій.

6. Myroshnychenko I., Kucherov D. Intellectual Control Systems for Aviation: Enhancing Autonomy and Safety. *Safetyin Aviation and Space Technologies. The Eleventh World Congress "AVIATION IN THE XXI-st CENTURY":* тези доп., 25-27 вер., 2024. Київ, Україна. С. 2.3.65-2.3.67 URL: https://jrnl.nau.edu.ua/index.php/congress/article/view/19225

Особистий внесок автора: розроблення концепції інтелектуальних систем керування для авіації, створення математичних моделей автономних алгоритмів, проектування GNN-архітектури CTDE, проведення симуляційних експериментів, аналіз безпеки та ефективності систем.

Особистий внесок Kucherov D.: формування загальної наукової концепції, наукове керівництво дослідженням, консультації щодо методології моделювання.

7. Kucherov D., Myroshnychenko I., Khalimon N., Tkachenko V. Training of the Dynamic Systems Control: A Neural Network or a Learning Algorithm.

Artificial Intelligence and Applications. 2025. DOI: https://doi.org/10.47852/bonviewAIA52025435

Особистий внесок автора.: формалізація задачі керування динамічною системою, участь у програмній реалізації моделювання та аналізі результатів.

Особистий внесок Kucherov D.: постановка наукової задачі, розробка методології дослідження, редагування рукопису.

Особистий внесок Khalimon N.: математичне обґрунтування алгоритмічних рішень, порівняння ефективності різних підходів керування.

Особистий внесок Tkachenko V.: побудова експериментальних сценаріїв, розробка функції «гістерезис» для компенсації перешкод, верифікація результатів моделювання.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ……. 18
ВСТУП…….. 20
РОЗДІЛ 1 АНАЛІЗ СТАНУ ПРОБЛЕМИ КООРДИНАЦІЇ ГРУПИ БПЛА 26
1.1 Актуальність дослідження децентралізованого управління групою БПЛА…….. 26
1.2 Класифікація підходів до координації групи БПЛА……..…….. 30
1.3 Огляд нормативно-правової бази для групових систем БПЛА….. 37
1.4 Аналіз існуючих методів координації групи БПЛА…..……... 39
1.5 Постановка задачі дослідження…..…… 55
Висновки до розділу 1 ……... 59
РОЗДІЛ 2 ТЕОРЕТИКО - МАТЕМАТИЧНІ МОДЕЛІ ТА МЕТОДИ ДЕЦЕНТРАЛІЗОВАНОГО УПРАВЛІННЯ РОЄМ БПЛА……..……. 63
2.1 Математична модель багатоагентної системи БПЛА як динамічного графа..... 63
2.2 Протокол передачі повідомлень між агентами …..…….. 69
2.3 Формальна постановка задачі …… 71
2.4 Архітектура мережі уваги на графах ……..……... 75
2.5 Механізм гарантування безпеки між агентами…… 80
2.6 Алгоритм навчання мультиагентній адаптації алгоритму PPO …. 82
2.7 Модель збурень та рандомізація домену……... 86
2.8 Методологія верифікації та метрики якості 88
Висновки до розділу 2 ……. 90
РОЗДІЛ 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ СИСТЕМИ ДЕЦЕНТРАЛІЗОВАНОГО УПРАВЛІННЯ РОЄМ БПЛА …… 92
3.1 Загальна архітектура програмного прототипу……. 92
3.2 Мультиагентне симуляційне середовище Gazebo…… 94
3.3 Реалізація актор уваги на графі та критик ……. 97
3.4 Реалізація мультеагенного алгоритму навчання з підкріпленням... 103

3.5 Реалізація міжагентного рівня безпеки ……. 106
3.6 Аналіз впливу компонентів моделі та оцінка її масштабованості... 108
3.7 Розгортання актор-мережі на бортовому ARM-контролері 112
3.8 Забезпечення відтворюваності результатів та автоматизація безперервної інтеграції і розгортання……. 116
Висновки до розділу 3 ……. 120
РОЗДІЛ 4 ВЕРИФІКАЦІЯ СИСТЕМИ ДЕЦЕНТРАЛІЗОВАНОГО УПРАВЛІННЯ РОЄМ БПЛА……... 123
4.1 Програма та структура верифікаційних експериментів……... 126
4.2 Апаратне та програмне забезпечення верифікації…… 127
4.3 Порівняльний аналіз методів координації …… 128
4.4 Абляційний аналіз та статистична достовірність…….. 134
4.5 Аналіз обчислювальних витрат та реального часу……... 136
Висновки до розділу 4 ……. 137
ВИСНОВКИ …….. 139
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ …… 143
ДОДАТКИ……..... 155
ДОДАТОК А Акти впровадження результатів дисертаційного дослідження…….. 155
ДОДАТОК Б Реалізація Graph Attention Actor (GATActor) мовою Python 158
ДОДАТОК В Конфігурація гіперпараметрів навчання MAPPO (YAML).. 162
ДОДАТОК Г Реалізація міжагентного QP Safety Layer…… 165
ДОДАТОК Д СПИСОК ПРАЦЬ ЗДОБУВАЧА…… 169

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

БПЛА безпілотне повітряне судно

НДР науково-дослідна робота

НМУ нейромережева модель управління

APF метод штучних потенційних полів (Artificial Potential Fields)

CBF бар'єрна функція управління (Control Barrier Function)

CNN згорткова нейронна мережа (Convolutional Neural Network)

CR_{inter} частота зіткнень між агентами рою (Inter-agent Collision Rate)

CTDE централізоване навчання з децентралізованим виконанням (Centralized Training, Decentralized Execution)

DAA виявлення та уникнення (Detect and Avoid)

Dec-POMDP децентралізована частково спостережувана марківська задача прийняття рішень

DQN глибока Q-мережа (Deep Q-Network)

EASA Агентство авіаційної безпеки Європейського Союзу

EI індекс масштабованості (Efficiency Index, EI = SR/N)

EKF розширений фільтр Калмана (Extended Kalman Filter)

GAT графова мережа уваги (Graph Attention Network)

GCN графова згорткова мережа (Graph Convolutional Network)

GNN графова нейронна мережа (Graph Neural Network)

GNSS глобальна навігаційна супутникова система

GRU рекурентна одиниця із затворами (Gated Recurrent Unit)

ICAO Міжнародна організація цивільної авіації

IMU інерційний вимірювальний блок (Inertial Measurement Unit)

LSTM довга короткострокова пам'ять (Long Short-Term Memory)

LQR лінійний квадратичний регулятор (Linear Quadratic Regulator)

MADDPG Multi-Agent Deep Deterministic Policy Gradient -мультиагентний алгоритм навчання з підкріпленням

MARL мультиагентне навчання з підкріпленням (Multi-Agent Reinforcement Learning)

MAPPO Multi-Agent Proximal Policy Optimization -мультиагентний алгоритм PPO

MLP багатошаровий персептрон (Multi-Layer Perceptron)

MPC прогнозне керування на основі моделі (Model Predictive Control)

PID пропорційно-інтегрально-диференційний регулятор

PPO Proximal Policy Optimization -алгоритм навчання з підкріпленням

QMIX алгоритм мультиагентного навчання на основі монотонної факторизації Q-функції

RL навчання з підкріпленням (Reinforcement Learning)

ROS операційна система для роботів (Robot Operating System)

R_{comm} радіус комунікації між агентами рою

SAR пошук і рятування (Search And Rescue)

SORA оцінка ризику для специфічних операцій БПЛА (Specific Operations Risk Assessment)

SR частота успішних місій (Success Rate)

SITL симуляція в контурі програмного забезпечення (Software In The Loop)

T_{form} час збору формації

UTM система управління трафіком БПЛА (UAV Traffic Management)

Δf середня похибка формації (відхилення позиції агентів від цільових точок), м

d_{safe} мінімально допустима безпечна відстань між агентами рою (2-3 корпуси апарата)

k кількість найближчих сусідів у графі зв'язку (k-nearest neighbors)

G(t) динамічний граф взаємодій рою в момент часу t

ВСТУП

Актуальність теми. Стрімкий розвиток безпілотних повітряних суден (БПЛА) та їх масове впровадження у цивільній, гуманітарній і оборонній сферах зумовив якісний перехід від застосування одиночних апаратів до координованих груп / роїв БПЛА. Задачі охоплення великих площ при моніторингу, пошуково-рятувальних операціях, картографуванні та точному землеробстві ефективно вирішуються лише груповим застосуванням апаратів. Практика свідчить, що для якісного моніторингу великих площ доцільним є одночасне використання групи з 5 - 20 безпілотних апаратів, які працюють у координації між собою. Такий підхід забезпечує суттєве підвищення продуктивності всієї системи і як свіжчать дослідження, швидкість оброблення та покриття території може зростати у 4-8 разів порівняно із застосуванням одного БПЛА за умов однакового обсягу контрольованої площі[1, 2].

У сучасних умовах застосування одного безпілотного літального апарата не завжди забезпечує необхідну швидкість збору даних і рівень покриття території. У зв’язку з цим особливого значення набуває використання груп безпілотних літальних апаратів, що дозволяє значно підвищити ефективність виконання відповідних завдань.

Масштабне впровадження групових систем безпілотних літальних апаратів в інфраструктурних та оборонних проєктах виступає важливим напрямом розвитку для України в умовах воєнного стану.

Основною науково-технічною проблемою при побудові групових систем є вибір парадигми управління. Використання єдиного контролера для керування всіма агентами робить систему вразливою: будь-який збій у центральному вузлі паралізує весь рій. Крім того, такий підхід не справляється з вимогами реального часу, коли кількість агентів перевищує п'ять – затримка між передачею стану та отриманням керуючих команд зростає понад 200 мс. За умов польоту на швидкостях від 5 до 15 м/с подібна затримка є неприйнятною для навігаційних алгоритмів і може призвести до критичних помилок управління [3].

Децентралізоване управління, де кожен БПЛА є автономним агентом з власною системою прийняття рішень на основі локальних спостережень та обміну повідомленнями із сусідами, є концептуально правильним, але породжує три взаємопов'язаних виклики:

- забезпечення точності утримання заданої геометричної формації ($\Delta f \leq$ 0,20 м);

- гарантія безпечних відстаней між апаратами без централізованого планування траєкторій (d_{safe} = 2-3 корпуси);

- масштабованість без перенавчання при зміні кількості агентів N у групі.

Існуючі підходи вирішують ці виклики лише частково. Наприклад, правила Рейнольдса (boids-модель) масштабуються, але забезпечують Δf = 0,45-0,80 м і не гарантують безпечних відстаней.

Метод потенційних полів (APF) дає механізм уникнення зіткнень, формуючи віртуальне поле, в якому ціль виступає джерелом притягувального потенціалу, а перешкоди – відштовхувального, але при цьому страждає від локальних мінімумів і O(N^2)-складності. Такий підхід дозволяє безперервно коригувати траєкторію руху апарата або групи апаратів у реальному часі.

Алгоритми мультиагентного навчання з підкріпленням MADDPG та QMIX навчаються для фіксованого N і не масштабуються без повторного навчання.

Сучасні методи на основі графових нейронних мереж (GNN) моделюють структуру рою природно, як динамічний граф $G(t)$ взаємодій, і принципово вирішують проблему масштабованості завдяки інваріантний до перестановок архітектурі та лінійній складності $O(N*k)$, де k – фіксована кількість сусідів. Однак існуючі GNN-реалізації тестуються переважно у 2D-просторі, не містять явних механізмів гарантування безпечних відстаней між агентами та не мають комплексної верифікації масштабованості.

Таким чином, актуальність дисертаційного дослідження зумовлена відсутністю ефективних децентралізованих нейромережевих методів управління роєм безпілотних літальних апаратів, здатних одночасно забезпечувати високу точність формаційного руху ($\Delta f \leq$ 0,20 м), низьку ймовірність міжагентних

зіткнень (CR_{inter} < 2 %) та масштабованість системи в діапазоні від N = 3 до N = 10 апаратів без необхідності додаткового перенавчання моделі в умовах реалістичного тривимірного середовища. Науковий інтерес до цієї проблеми зумовлений також перспективністю інтеграції графових нейронних мереж, зокрема архітектури Graph Attention Network (GAT), з алгоритмами багатоагентного підкріплювального навчання, такими як Multi-Agent Proximal Policy Optimization (MAPPO), у межах парадигми централізованого навчання з децентралізованим виконанням (CTDE). Зазначений підхід розглядається як теоретично обґрунтований напрям розв’язання задачі одночасного забезпечення точності формаційного керування, безпечної взаємодії агентів та масштабованості ройових систем БПЛА.

Зв'язок роботи з науковими програмами, планами, темами. Дисертаційне дослідження виконано на кафедрі комп'ютерних інформаційних технологій факультету комп’ютерних наук та технологій Національного університету «Київський авіаційний інститут» у рамках двох науково-дослідних робіт: НДР 26-2024/14.03 «Підвищення достовірності цифрової обробки зображень з бортової відеокамери БПЛА» (2024-2025) - дослідження методів бортової обробки даних та алгоритмів навігації для БПЛА; НДР 17-2022/09.01.05 «Інтелектуальна система управління локальною електромережею класу Microgrid на основі SmartGrid» (0122U200903, 2022-2024) - розробка методів розподіленого управління мультиагентними системами з використанням GNN та MARL. Принципи децентралізованого управління складними системами, розроблені в рамках другої НДР, перенесено та адаптовано для задачі координації рою БПЛА.

Мета і завдання дослідження. Метою дослідження є підвищення ефективності групового управління БПЛА на основі застосування верифікованого децентралізованого нейромережевого управління.

Для досягнення поставленої мети визначено такі завдання дослідження:

1) виконати систематичний огляд та порівняльний аналіз існуючих методів координації групи БПЛА і виявити їхні обмеження щодо точності, безпеки та масштабованості;

2) розробити математичну модель N-агентної системи БПЛА на основі графового підходу- із забезпеченням безпечної взаємодії агентів ;

3) запропонувати архітектуру нейромережевого управління групою БПЛА;

4) розробити механізми гарантування безпеки, точності та масштабування групи агентів та провести їх експериметральне тестування;

Об'єкт дослідження є процес децентралізованого управління групою автономних БПЛА в умовах обмеженого зв'язку, динамічної топології взаємодій та вимог безпеки між агентами рою.

Предмет дослідження графові нейронні мережі та методи мультиагентного навчання з підкріпленням для координації групи БПЛА.

Методи дослідження. В процесі дослідження використовувались методи математичне моделювання динаміки мультиагентних систем і теорія динамічних графів – для побудови формальної моделі рою; методи глибокого навчання (GNN, GAT, MLP) – для синтезу нейромережевих компонентів архітектури; алгоритми мультиагентного навчання з підкріпленням (MAPPO, MADDPG) – для навчання агентів у симуляційному середовищі; метод потенційних полів – для синтезу функції безпеки між агентами; симуляційне моделювання (Gazebo 11, ROS Noetic, ArduPilot SITL, PyTorch + PyG) – для верифікації результатів; статистичний аналіз (критерій Вілкоксона, 95% довірчі інтервали) – для підтвердження вірогідності результатів.

Наукова новизна одержаних результатів. У результаті дослідження одержано такі нові наукові результати:

вперше розроблено метод децентралізованого просторового управління роєм БПЛА, який, за рахунок застосування графової нейронної мережі з механізмом уваги на основі парадигми централізованого навчання та децентралізованого виконання з інваріантним до перестановок агрегуванням

повідомлень між агентами, що замінює конкатенацію глобальних станів та забезпечує обчислювальну лінійну складність актора замість квадратичної, що дозволяє досягти масштабованості рою БПЛА без перенавчання за одночасного підвищення точності утримання формації і часу бортового виконання на ARM-процесорі;

удосконалено метод мультиагентного навчання з підкріпленням, який за рахунок введення «Актора» та «Критика» уваги на графі з глобальним об'єднанням замість конкатенації станів агентів, що дозволяє знизити обчислювальну складність алгоритму з квадратичної до лінійної та забезпечити стійкість навчання при зміні кількості агентів у рої БПЛА;

удосконалено підхід до забезпечення безпечної взаємодії агентів рою БПЛА за рахунок впровадження дворівневого механізму гарантування безпеки, який поєднує формування функції винагороди з урахуванням міжагентних відстаней та корекцію керуючих дій на основі квадратичного програмування, що дозволяє забезпечити дотримання безпечних дистанцій між агентами та знизити ризик міжагентних зіткнень під час виконання групових місій;

набуло подальшого розвитку метрики оцінювання масштабованості алгоритмів координації груп БПЛА, які, за рахунок уніфікованого формування на основі частоти успішних місій та набору тестових сценаріїв з нормуванням (центруванням) до еталону, дозволяють об'єктивно порівнювати робастність різних алгоритмів координації за змінної кількості агентів.

Практичне значення одержаних результатів. Розроблено програмний прототип моделі ройового управління (~4 000 рядків Python, PyTorch 2.0, PyG 2.3, ROS Noetic) з програмним пакетом підтримки розрахунків для середовищ сумісних з мовою Python та MAVLink-інтерфейсом до програмної системи ArduPilot, який придатний до застосування у системах координованого картографування, що дозволяє підвищити покриття у 8-10 разів краще за одиночний апарат при виконанні пошуково-рятувальних операцій та оборонних БПЛА-місіях.

Проведені в дисертаційній роботі дослідження реалізовані й впроваджені (акти подані у Додатку А до дисертаційної роботи): у навчальний процес кафедри комп'ютерних систем та мереж КАІ (Акт впровадження результатів від 2.03.2026 року) та в практичну діяльність підприємств: ТОВ «ДЕФІНДЕР ДІНАМІКС» (Акт впровадження результатів від 4.03.2026 року), ПрАТ «РАМЗАЙ» (Довідка про впровадження результатів)

Особистий внесок здобувача. Усі наукові результати, що виносяться на захист, отримано здобувачем самостійно. Постановку задачі та визначення напряму дослідження здійснено спільно з науковим керівником д.т.н., проф. Кучеровим Д. П. У спільних публікаціях здобувачеві належить: розробка математичної моделі рою як динамічного графа, проектування GNN-архітектури CTDE, реалізація та верифікація програмного прототипу, проведення симуляційних експериментів та статистичний аналіз результатів.

Апробація результатів дисертації. Основні результати дисертаційного дослідження доповідалися та обговорювалися на таких наукових заходах:

- CEUR Workshop Proceedings, 2023;
- Міжнародна науково-технічна конференція «АВІА-2023», Київ, НАУ, квітень 2023;
- 13th International Conference on Dependable Systems, Services and Technologies (DESSERT), Афіни, Греція, жовтень 2023;
- 2023 IEEE 7th International Conference on Methods and Systems of Navigation and Motion Control (MSNMC), Київ, жовтень 2023;
- 2023 IEEE 4th KhPI Week on Advanced Technology (KhPIWeek), Харків, жовтень 2023;
- AVIATION IN THE XXI-st CENTURY: Safety in Aviation and Space Technologies. The Eleventh World Congress, Київ, вересень 2024.

Публікації. За темою дисертаційного дослідження опубліковано 13 наукових праць, у тому числі: 2 публікацій у наукових виданнях, що індексуються у міжнародній наукометричній базі Scopus; 4 статей у фахових

виданнях України категорії Б; 1 стаття у міжнародному журналі; 6 тез доповідей на міжнародних та всеукраїнських конференціях.

Структура та обсяг дисертації. Дисертація складається з анотації, вступу, 4 розділів, висновків і списку використаних джерел. Повний обсяг дисертації становить 172 сторінок, із них – 155 основного тексту. Робота містить 17 рисунків, 24 таблиці, 5 додатків. Список використаних джерел налічує 101 найменування.

РОЗДІЛ 1 АНАЛІЗ СТАНУ ПРОБЛЕМИ КООРДИНАЦІЇ ГРУПИ БПЛА

1.1 Актуальність дослідження децентралізованого управління групою БПЛА

Розвиток технологій безпілотних повітряних суден (БПЛА) упродовж останніх років суттєво розширив можливості їх використання у цивільних, гуманітарних та оборонних завданнях. Однією з ключових тенденцій сучасного етапу розвитку безпілотних систем є перехід від експлуатації окремих апаратів до застосування груп БПЛА, що функціонують у режимі узгодженої взаємодії. Такий підхід дозволяє значно підвищити ефективність виконання складних завдань, пов’язаних із моніторингом великих територій, проведенням пошуково-рятувальних операцій, картографуванням місцевості та виконанням робіт у сфері аграрного виробництва.

Практика використання безпілотних систем свідчить, що одночасне застосування групи апаратів (орієнтовно 5–20 одиниць) забезпечує значно вищу продуктивність і швидкість охоплення території порівняно з використанням одного безпілотного апарата, оскільки завдання можуть виконуватися паралельно кількома платформами [1, 2]. З огляду на це особливої актуальності набувають питання організації ефективного управління такими багатоагентними системами.

До перспективних напрямів застосування груп БПЛА також належить інспектування об’єктів транспортної інфраструктури (мости, автомобільні магістралі, інженерні споруди), де використання БПЛА дозволяє підвищити швидкість збору даних, точність візуального контролю та ефективність оброблення зображень для задач кластеризації й розпізнавання дефектів [1**]** У зв’язку з цим дослідження методів ефективного управління групами БПЛА, зокрема на основі інтелектуальних алгоритмів, набуває важливого наукового та прикладного значення.

Глобальний ринок роїв безпілотних повітряних суден (БПЛА) демонструє високі темпи зростання. За даними аналітичного агентства Grand View Research (2023), сегмент систем із багатьма апаратами (multi-UAV) у 2022 році оцінювався приблизно в 4,2 млрд дол., а до 2030 року очікується його збільшення до 21,8 млрд дол. при середньорічному темпі зростання (CAGR) 22,7%. В рамках концепції U-Space, розробленої EASA та SESAR, передбачається, що до 2035 року у повітряному просторі Європейського Союзу одночасно може функціонувати до 400 000 БПЛА на добу, при цьому значна частка польотів буде виконуватися у вигляді координаційних групових місій [3, 4].

Для України розвиток і застосування групових безпілотних повітряних суден набуває особливої стратегічної значущості в умовах військового стану. Використання роїв дронів у розвідці, проведенні радіоелектронної боротьби та виконанні координованих тактичних операцій висуває принципово нові вимоги до алгоритмів управління, зокрема забезпечення високого рівня автономності, надійності функціонування при втраті зв’язку та гарантій безпеки взаємодії апаратів у складі рою. Проблема забезпечення безпеки у складних авіаційних системах охоплює не лише технічну надійність, а й ризики помилкового прийняття рішень. Для оцінювання таких ризиків застосовуються нейромережеві моделі, здатні враховувати часові нормативи, послідовність дій та механізми зворотного зв’язку. Водночас у цивільному секторі групові БПЛА знаходять широке застосування для виконання завдань, що потребують швидкого та ефективного охоплення територій: моніторинг руйнувань і оцінка матеріальних збитків (покриття площі 1 км² за 15 хвилин, що приблизно в 8 разів швидше порівняно з одиночним апаратом); проведення пошуково-рятувальних операцій у важкодоступних або небезпечних районах; координоване картографування мінних полів та заражених територій; а також підтримка аграрного виробництва при масштабних спростереженнях сільськогосподарських угідь [5].

Однією з ключових науково-технічних проблем при побудові та експлуатації групових систем безпілотних повітряних суден (БПЛА) є визначення оптимальної парадигми управління, що забезпечує ефективну

координацію агентів рою, високу надійність та адаптивність до динамічних умов середовища. У науковій та практичній літературі традиційно розглядають два основні підходи: централізоване та децентралізоване управління.

Централізоване управління передбачає наявність єдиного контролера, який обробляє глобальну інформацію про стан усіх агентів і формує команди для кожного апарата рою. Такий підхід дозволяє досягати високої глобальної оптимальності та координованого виконання завдань. Водночас централізована архітектура має низку суттєвих обмежень. По-перше, вона формує «єдину точку відмови»: у разі виходу з ладу центрального сервера або втрати каналу зв'язку вся група втрачає координацію, що суттєво підвищує ризики при виконанні швидкоплинних або критично важливих завдань. По-друге, затримки обміну даними між агентами та центральним контролером суттєво зростають із розміром рою. Практичні дослідження показують, що для групи з більш ніж п'яти апаратів затримка передачі стану N агентів на сервер та повернення команд до виконавців перевищує 200 мс, що перевищує поріг реального часу для авіаційних систем, особливо при швидкості польоту 5–15 м/с. Незначні на око зміни у стані системи, що відбуваються швидше за 200 мс, можуть мати критичне значення для безпеки та точності координації рою [6].

Децентралізоване управління передбачає, що кожен БПЛА функціонує як автономний агент, здатний приймати рішення на основі локальних спостережень та обміну повідомленнями із сусідніми апаратами. Такий підхід усуває обмеження централізованих систем, підвищуючи стійкість групи до відмов та забезпечуючи гнучкість у динамічному середовищі. Разом з тим децентралізоване управління породжує нові науково-технічні виклики, зокрема:

1. Глобальна узгодженість поведінки рою – як забезпечити координацію всієї групи при обмеженому локальному обміні інформацією;
2. Безпека польотів – як підтримувати безпечні міжагентні відстані без централізованого планування траєкторій;

3. Масштабованість системи – як гарантувати коректну роботу алгоритму при збільшенні кількості агентів (наприклад, з 5 до 10 БПЛА) без необхідності повторного навчання.

Саме ці три аспекти – узгодженість, безпечна відстань та масштабованість – формують основну наукову проблему. Еволюція підходів до вирішення зазначеної проблеми в науковій літературі останнього десятиліття свідчить про поступове ускладнення та підвищення ефективності алгоритмів координації роїв. На ранньому етапі використовувалися правила поведінки агентів Рейнольдса [7] та методи потенційних полів [8], які забезпечували базову узгодженість руху, проте не гарантували високу точність формацій та безпечні відстані між апаратами.

Подальший розвиток пов’язаний із використанням класичних мультиагентних підходів на основі навчання з підкріпленням (MADDPG, QMIX), які дозволяють формувати координацію через оптимізацію колективної функції винагороди, але масштабуються лише на фіксовану кількість агентів.

Сучасні методи на основі графових нейронних мереж (GNN) дозволяють природним чином моделювати топологічну структуру рою як динамічний граф взаємодій і потенційно забезпечують масштабованість, однак більшість існуючих підходів не містять явних механізмів гарантування безпечних міжагентних відстаней та точності формації на рівні $\Delta f < 0{,}2–0{,}3$ м [7-16].

Таким чином, актуальність проведеного дослідження зумовлюється нагальною практичною потребою у створенні масштабованих систем координації груп БПЛА у складі 5 - 20 апаратів, здатних ефективно виконувати завдання як у цивільному, так і в оборонному секторах. Існуючі методи не дозволяють одночасно забезпечити високий рівень масштабованості без перенавчання, гарантії безпечної взаємодії між агентами та точність формацій на рівні $\Delta f < 0{,}2$ м, що обмежує їх практичне застосування.

Перспективним напрямом у вирішенні цих проблем є використання підходів на основі графових нейронних мереж у поєднанні з парадигмою Centralized Training, Decentralized Execution (CTDE). Така концепція дозволяє

поєднувати переваги централізованого навчання з децентралізованим виконанням, забезпечуючи ефективну координацію рою, підвищену стійкість до відмов та адаптивність до динамічних умов середовища. Реалізація та вдосконалення подібних моделей є ключовою науковою задачею, що визначає напрямок і мету даного дослідження.

1.2 Класифікація підходів до координації групи БПЛА

1.2.1 Централізовані та децентралізовані системи управління

Системи управління групою БПЛА класифікують за ступенем централізації прийняття рішень. Централізована архітектура передбачає наявність єдиного планувального вузла (наземного або бортового мастер-агента), який отримує повний стан усіх агентів і формує команди для кожного з них. Обчислювальна складність такого підходу зростає як $O(N^2)$ або $O(N^3)$ залежно від методу оптимізації, що робить його непридатним для $N > 5$-7 у режимі реального часу. Надійність системи суттєво знижується при відмові центрального вузла або деградації каналу зв'язку [17, 18, 19].

Децентралізована архітектура (розподілена система агентів) означає, що кожен БПЛА є автономним агентом: він спостерігає власний стан, обмінюється повідомленнями лише з найближчими сусідами (в радіусі R_{comm}) та самостійно обчислює керуючу дію. Обчислювальна складність бортового алгоритму не залежить від загального N при правильно побудованій архітектурі -це є ключовою перевагою. Недоліком є ускладнена глобальна оптимізація: без знання повного стану системи локальні рішення можуть бути субоптимальними [20, 21].

Гібридна парадигма централізованого навчання/децентралізованого виконання CTDE (Centralized Training, Decentralized Execution) поєднує переваги централізованого та децентралізованого підходів до навчання мультиагентних систем. У рамках CTDE навчання відбувається централізовано, із доступом до глобальної інформації про стан усіх агентів, що дозволяє отримати узгоджені стратегії поведінки. Під час виконання, навпаки, кожен агент реалізує власну

політику на основі лише локальних спостережень, що забезпечує автономність, масштабованість та зниження вимог до комунікації. Такий підхід широко застосовується у завданнях кооперативної навігації та планування траєкторій для рою БПЛА, де централізоване навчання сприяє узгодженому навчанню агентів, а децентралізоване виконання дозволяє ефективно працювати в розподіленому середовищі без потреби обміну глобальним станом під час польоту [22, 23, 24].

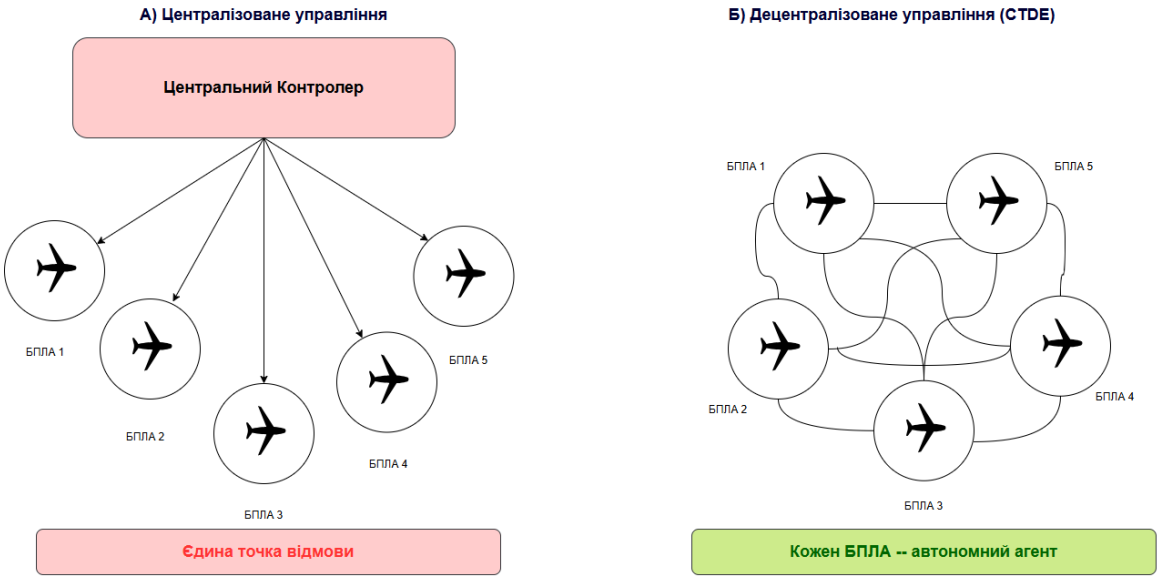

Рис. 1.1. Порівняння парадигм управління групою БПЛА

Отдже, вибір архітектури управління групою БПЛА визначає баланс між узгодженістю дій агентів, обчислювальною ефективністю та автономністю.

Зазначені архітектури управління (централізована, децентралізована та гібридна CTDE) реалізуються на основі різних класів алгоритмів керування та обробки інформації. Вибір конкретного алгоритмічного забезпечення безпосередньо впливає на ефективність, адаптивність та стійкість мультиагентної системи БПЛА [25].

У межах централізованих систем традиційно застосовуються класичні методи керування, такі як PID-регулятори, LQR/LQG та модельно-прогнозне керування (MPC), які забезпечують глобальну оптимізацію за умови наявності

повної інформації про стан системи. Проте їх ефективність знижується в умовах нелінійної динаміки та обмежених обчислювальних ресурсів.

Для децентралізованих систем більш характерними є розподілені та адаптивні алгоритми, зокрема методи ковзного режиму, адаптивного керування та алгоритми консенсусу, які дозволяють агентам приймати рішення на основі локальної інформації та взаємодії з сусідами. Такі підходи забезпечують високу масштабованість і відмовостійкість, проте можуть призводити до субоптимальних глобальних рішень.

Сучасні гібридні підходи, зокрема в рамках парадигми CTDE, активно використовують методи машинного навчання, включаючи глибокі нейронні мережі та навчання з підкріпленням (RL). Вони дозволяють формувати узгоджені стратегії поведінки агентів під час централізованого навчання та забезпечувати автономне прийняття рішень під час децентралізованого виконання. У сучасних мультиагентних системах БПЛА та НРЗ окрему роль відіграють механізми розподілу задач, управління ресурсами та координації маршрутів, що можуть реалізовуватися з використанням хмарних технологій та методів машинного навчання [2**].** Особливо ефективними є підходи мультиагентного навчання з підкріпленням (MARL), які забезпечують координацію рою БПЛА в умовах невизначеності та динамічного середовища [25].

У таблиці 1.1 наведно порівняння основних основних класів алгоритмів керування БПЛА

Аналіз наведених методів показує, що класичні алгоритми залишаються ефективними для централізованих систем із повною інформацією, тоді як децентралізовані архітектури потребують використання адаптивних та розподілених підходів. Водночас найбільш перспективними є гібридні інтелектуальні методи, які поєднують переваги класичних моделей і сучасних технологій машинного навчання, забезпечуючи баланс між обчислювальною ефективністю, автономністю та якістю прийняття рішень.

Таблиця 1.1

Класи алгоритмів керування БПЛА в контексті архітектур управління

Клас методів	Характеристика	Переваги	Недоліки
Класичні (PID, LQR)	Лінійні або лінеаризовані моделі	Простота, стабільність	Обмежена адаптивність
Оптимальні (MPC)	Оптимізація керування в реальному часі	Висока точність	Висока обчислювальна складність
Методи ковзного режиму	Робота в умовах невизначеності	Робастність	Дребготіння сигналу
Адаптивні методи	Зміна параметрів у реальному часі	Гнучкість	Складність реалізації
Нейромережеві	Навчання на даних	Робота з нелінійністю	Потреба в даних
RL / MARL	Навчання через взаємодію	Автономність, координація	Висока складність навчання

Джерело: узагальненно за [25]

Аналіз сучасних наукових джерел [26-31] свідчить, що розвиток систем керування безпілотними літальними апаратами ґрунтується на інтеграції класичних математичних моделей із сучасними методами обробки даних. Традиційні алгоритми керування, зокрема PID-регулятори, LQR, ADRC та методи ковзного режиму, і надалі відіграють базову роль, однак їх функціональні можливості істотно розширюються завдяки впровадженню нейромережевих і гібридних підходів. Зокрема, використання глибоких нейронних мереж забезпечує підвищення точності відтворення заданих траєкторій за умови наявності достатнього обсягу навчальних даних [25]. У складних авіаційних середовищах колективне прийняття рішень може здійснюватися за участі пілотів, диспетчерів та операторів БПЛА, що формує гібридну систему інтелектуального управління. Для таких задач використовуються інтелектуальні системи підтримки прийняття рішень, які поєднують штучний та природний інтелект і дозволяють працювати в умовах невизначеності

Разом із цим, задачі автономної навігації та планування маршрутів дедалі частіше розв’язуються із застосуванням інтелектуальних технологій. Методи машинного навчання, зокрема нейронні мережі, дозволяють безпілотним апаратам оперативно коригувати траєкторію руху з урахуванням змін

середовища та появи нових перешкод. Особливу роль відіграє навчання з підкріпленням, яке дає змогу ройовим системам самостійно формувати стратегії уникнення зіткнень і узгоджувати групову поведінку. При цьому важливими залишаються задачі підвищення енергоефективності та синхронізації дій великої кількості агентів у межах спільних місій. Координація ройових систем забезпечується за рахунок децентралізованих алгоритмів і протоколів обміну інформацією, що сприяють масштабованості та підвищенню надійності функціонування. Використання методів штучного інтелекту дозволяє таким системам демонструвати високий рівень відмовостійкості та здатність до адаптації у випадку втрати окремих елементів. Загалом, поєднання сучасних підходів до обробки інформації, включаючи нейромережеві моделі, алгоритми оцінювання стану та інтелектуальні методи планування, формує передумови для створення високоефективних автономних систем, що характеризуються підвищеною точністю, безпечністю та результативністю виконання польотних завдань [25].

Отже, проведений аналіз засвідчує, що ефективність управління групою БПЛА визначається узгодженим поєднанням архітектури системи та обраного класу алгоритмів керування. Централізовані підходи орієнтовані на застосування класичних і оптимізаційних методів за умов повної інформації, тоді як децентралізовані системи базуються на розподілених, адаптивних та консенсусних алгоритмах, що забезпечують масштабованість і відмовостійкість. Гібридні архітектури, зокрема в межах парадигми CTDE, інтегрують методи машинного навчання та мультиагентного навчання з підкріпленням, формуючи узгоджені стратегії поведінки агентів.

1.2.2 Формальна модель рою як динамічного графа

Природною математичною моделлю рою БПЛА є динамічний граф $G(t) = (V, E(t))$, де $V = \{1, ..., N\}$ – множина агентів-вершин, а $E(t) \subseteq V * V$ –

множина ребер зв'язку. Ребро $(i, j) \in E(t)$ існує тоді і лише тоді, коли відстань між агентами i та j не перевищує радіус комунікації R_{comm}:

$$(i, j) \in E(t) \leftrightarrow \|p_i(t) - p_j(t)\| \leq R_{comm} \quad (1.1)$$

де $p_i(t) \in R^3$ – вектор положення агента i у момент часу t.

Така модель природно відображає два ключових аспекти реальних систем 32, 33, 34, 35]:

1. Динамічна топологія зв'язку – агенти можуть входити та виходити з радіуса зв'язку один одного;

2. Можлива незв'язність графа – при достатньо великих відстанях між підгрупами агентів граф може розпадатися на кілька компонентів/

Для задачі утримання формації вводиться цільова конфігурація агента i_i :

$$p_i^* = p_{centroid}(t) + \delta_i, \quad (1.2)$$

$P_{centroid}(t)$ – центр формації, δ_i –заданий зсув і-го агента.

Похибка формації агента і визначається як:

$$\Delta f_i(t) = \|p_i(t) - p_i^*(t)\|^2 \quad (1.3)$$

Середня похибка формації по всій групі обчислюється за формулою:

$$\Delta f(t) = \frac{1}{N}\sum_{i=1}^{N} \Delta f_i(t), \quad (1.4)$$

Для забезпечення безпеки польоту вводиться умова мінімальної відстані між агентами:

$$\left\|p_i(t) - p_j\ (t)\right\| \quad \geq d_{safe,} \quad i \neq j, \qquad t \in [0, T], \quad (1.5)$$

де d_{safe} – мінімально допустима відстань між апаратами. У авіаційній практиці вона приймається на рівні 2–3 корпусів літального апарата, що відповідає міжнародним стандартам безпеки групових польотів [36, 37, 38, 39, 40].

Українські дослідження також підтверджують застосування графових моделей та алгоритмів контролю відстані у мультиагентних системах БПЛА,

включаючи розподілене управління, узгодження формації та безпекові умови [37].

Таким чином, описана модель дозволяє формалізувати поведінку рою БПЛА через динамічний граф, враховувати зміни топології зв'язку, контролювати похибку формації та забезпечувати безпеку польоту, що є основою для подальшого розроблення алгоритмів кооперативного управління та планування траєкторій

1.2.3 Метрики оцінювання ефективності групового управління

Для об'єктивної оцінки ефективності алгоритмів групового управління роєм безпілотних літальних апаратів використовуються числові метрики, що відображають як якість утримання формації, так і безпеку та здатність алгоритмів масштабуватися при зміні кількості агентів.

Однією з основних метрик є похибка формації Δf *(м)* [34, 39], яка визначає середнє відхилення фактичних позицій агентів від їх цільових позицій у заданій конфігурації. Вона служить мірою точності узгодженого руху групи і може бути формалізована як середнє арифметичне модулів індивідуальних похибок агента від відповідної цільової точки.

Для оцінювання безпеки польоту вводиться частота зіткнень між агентами CR_{inter} (%), що визначається як відсоток часових кроків моделювання або польоту, протягом яких хоча б одна пара агентів порушує умову мінімальної безпекової відстані d_{safe}. Ця метрика відображає здатність алгоритму забезпечувати уникнення ризикових ситуацій при реальному русі рою [39].

Іншою важливою характеристикою є частота успішних місій *SR* (%), яка показує відсоток польотних епізодів, у яких усі агенти досягли заданої формації без зіткнень та зі збереженням зв'язності протягом польоту [35]. Ця метрика дозволяє оцінити загальну здатність алгоритму до координації в різних сценаріях групової взаємодії.

Ще одна кількісна оцінка – час збору формації T_{form} *(с)*, що визначається як тривалість від старту або команди на перестроювання до моменту, коли похибка формації Δf стає меншою за заданий поріг $\Delta f_{threshold}$. Час T_{form} характеризує динамічну швидкість адаптації групи до цільової конфігурації [34].

Для аналізу масштабованості алгоритмів використовують індекс ефективності масштабування [41]:

$$EI = \frac{SR}{N}, \tag{1.6}$$

де N – кількість агентів у рої.

Така нормалізація дозволяє коректно порівнювати ефективність між алгоритмами, що працюють з різними кількостями агентів, і враховує зниження ефективності при збільшенні розмірності мультиагентної системи [41].

Застосування зазначених метрик забезпечує комплексну оцінку алгоритмів групового управління за параметрами точності, безпеки, надійності виконання задач та здатності до масштабування, що є критично важливими аспектами при розробці практично застосовних рішень для розподілених мультиагентних систем БПЛА.

1.3 Огляд нормативно-правова база для групових систем БПЛА

Перед експлуатацією групових БПЛА необхідно враховувати нормативно-правові акти та міжнародні стандарти, що визначають вимоги до безпеки, надійності та інтеграції у повітряний простір. Регуляторне середовище встановлює вимоги до технічної надійності, верифікації та безпеки систем групових БПЛА. Міжнародна організація цивільної авіації (ICAO) у своєму документі [42] окреслює основні принципи інтеграції безпілотних літальних апаратів у загальний повітряний простір . У контексті координації групових операцій ключове значення набуває концепція управління трафіком БПЛА, яка передбачає організацію динамічних маршрутів і висотних ешелонів для роїв

БПЛА, подібно до традиційного диспетчерського контролю за пілотованими літаками.

Агентство авіаційної безпеки ЄС (EASA) у регламентах (ЄС) № 2019/945 [43] та № 2019/947 [44] класифікує операції БПЛА за рівнем ризику. Зокрема, групові польоти безпілотних літальних апаратів зазвичай відносять до категорії «Specific», що передбачає проведення попередньої оцінки ризиків за методологією SORA (Specific Operations Risk Assessment). Однією з ключових вимог для таких операцій є наявність процедур запобігання зіткненням як між БПЛА в рої, так і між роєм та пілотованими літаками (Detect and Avoid, DAA).

Стандарт ASTM F3269-17 [45] встановлює методи безпечного обмеження поведінки безпілотних літальних систем у польоті в межах безпекового «конверту». У запропонованій архітектурі компонент, що відповідає за збереження безпечної відстані, узгоджується з цими підходами. А в стандарті EUROCAE ED-269 [46] визначено мінімальні експлуатаційні вимоги до систем управління трафіком БПЛА, включно з координацією групових польотів.

В Україні діяльність групових безпілотних літальних апаратів регламентується положеннями Повітряного кодексу України [47], що встановлює правові основи використання повітряного простору, включно із визначенням безпілотних повітряних суден та принципами авіаційної безпеки. Положення про використання повітряного простору України (Постанова КМУ № 954 від 22.11.2017) [48] деталізує порядок регулювання польотів та забезпечення безпеки повітряного руху. Для критичної інфраструктури застосовуються положення Закону України № 1882-IX від 16.11.2022 [49], що регламентують захист об’єктів, у тому числі від ризиків, пов’язаних із БПЛА.

Ця нормативно-правова база забезпечує законність, безпеку та ефективну інтеграцію групових БПЛА у повітряний простір. Виконання вимог цих документів гарантує контроль ризиків і безпечну експлуатацію групових безпілотних систем.

1.4 Аналіз існуючих методів координації групи БПЛА

1.4.1 Правила Рейнольдса (boids-модель)

Класичний підхід до моделювання зграйної поведінки, запропонований Крейгом Рейнольдсом у 1987 році, базується на трьох основних локальних правилах – сепарація, вирівнювання та згуртованість – що дозволяють кожному агенту самостійно координувати рух, використовуючи лише інформацію про найближчих сусідів у межах певного радіуса сприйняття. Такий підхід імітує злагоджену поведінку групи без централізованого контролю, демонструючи, як прості локальні взаємодії призводять до складної глобальної координац [7].

Модель Рейнольдса вирізняється високою обчислювальною ефективністю, зі складністю $O(N*k)$, де N – загальна кількість агентів в системи, а k – кількість сусідів у межах радіуса сприйняття, що дозволяє природно масштабувати алгоритм на довільне число агентів та забезпечує повну децентралізацію управління [7]. Водночас цей класичний підхід має принципові обмеження для практичного використання у системах управління БПЛА. Зокрема, він не передбачає механізму підтримки заданої геометричної конфігурації (формації) та не гарантує безпечні відстані між апаратами, оскільки ваги локальних правил підбираються евристично і можуть не забезпечувати мінімально допустиму відстань між апаратами при зміні швидкостей [50, 51]. Крім того, похибка формації Δf при застосуванні правил Рейнольдса зазвичай становить 0,45–0,80 м, тоді як, наприклад, для задач точного картографування та сільськогосподарського землеробства потрібна точність $\Delta f \leq 0{,}20$ м [52].

1.4.2 Метод потенційних полів

Метод потенційних полів (Artificial Potential Fields, APF) базується на тому, що кожен агент рухається під дією двох типів потенціалів [51, 53]:

- притягувальний потенціал (U_{att}), орієнтує агента на цільову точку формації,

- відштовхувальний потенціал (U_{rep}), запобігає зіткненням з іншими агентами та перешкодами.

Загальна управляюча сила для *i*-го агента обчислюється як:

$$F_t = -\nabla U_{att}(p_i) + \sum \nabla U_{rep}(p_i, p_j) \quad (1.7)$$

Відштовхувальний потенціал зазвичай задається формулою:

$$U_{rep}(p_i, p_j) = \begin{cases} \frac{\eta}{\|p_i - p_j\|^2}, & \|p_i - p_j\| \le R_{rep} \\ 0, \textit{в іншому випадку} \end{cases} \quad (1.8)$$

де p_i,p_j – позиції агентів i та j,

η – коефіцієнт інтенсивності відштовхування,

R_{rep} – радіус дії відштовхувального потенціалу.

Перевагою методу є його обчислювальна простота та можливість застосування в реальному часі, що робить його ефективним для задач навігації та координації рою безпілотних апаратів. Водночас до основних недоліків належать наявність локальних мінімумів потенціального поля (агент «застряє» між рівними по силі потенціалами), можливість осциляцій у вузьких проходах та складність налаштування параметрів у динамічних середовищах, відсутність механізму узгодженого утримання заданої формації (похибка формації при чистому APFдорівнює 0,38-0,65 м), обчислювальна складність $O(N^2)$ [54, 55, 56].

Але з іншого боку, метод потенційних полів забезпечує природний механізм уникнення зіткнень через відштовхувальне поле – саме цей принцип інтегрований у запропоновану архітектуру як компонент функції безпеки.

Таким чином, метод потенційних полів є базовим інструментом побудови алгоритмів децентралізованого управління мультиагентними системами, зокрема роями БПЛА, та часто використовується як основа або складова більш складних гібридних підходів. Завдяки поєднанню простоти реалізації, децентралізованого характеру управління та ефективного механізму уникнення зіткнень метод потенційних полів залишається одним із найбільш поширених підходів при проєктуванні систем групового керування безпілотними літальними апаратами.

1.4.3 Мультиагентне навчання з підкріпленням

Методи мультиагентного навчання з підкріпленням (Multi-Agent Reinforcement Learning, MARL) формалізують задачу координації рою безпілотних апаратів у вигляді стохастичної гри Маркова або частково спостережуваного процесу прийняття рішень із децентралізованим керуванням (Decentralized Partially Observable Markov Decision Process, Dec-POMDP) [57, 58, 59].

У межах такої постановки кожен агент ($i \in I$) обирає дію ($a_i \in A$) на основі власного локального спостереження ($o_i \in O$) з метою максимізації очікуваної кумулятивної винагороди. Спільна задача мультиагентної взаємодії формально задається [60; 61]:

$$G = \langle I, S, A, P, R, O, \gamma \rangle \quad (1.9)$$

де I – множина агентів {1, ..., N},

S – глобальний простір станів середовища,

A – простір спільних дій,

P – функція переходу ($S * A \rightarrow \Delta(S)$);

R – функції винагороди,

O – функції спостереження,

γ – коефіцієнт дисконтування.

MADDPG (Multi-Agent DDPG) запропонована [15] є одним із базових підходів багатоагентного навчання з підкріпленням, реалізованим у парадигмі централізованого навчання та децентралізованого виконання (CTDE). У межах цього підходу кожен агент має власну актор-мережу (π_i), що визначає політику дій, та критик-мережу (Q_i) яка оцінює якість прийнятих рішень. Під час навчання критик використовує розширений вхідний простір, що включає глобальну інформацію про стани та дії всіх агентів ($o_1, ..., o_N, a_1, ..., a_N$) що дозволяє зменшити вплив нестаціонарності середовища, характерної для багатоагентних систем. Водночас на етапі виконання кожен агент приймає рішення виключно на

основі локального спостереження (o_i), що забезпечує децентралізованість управління [62].

Разом з тим, даний підхід має низку обмежень, що ускладнюють його застосування в задачах управління роями БПЛА. Зокрема, розмір вхідного простору критика зростає квадратично відносно кількості агентів, тобто має складність ($O(N^2)$), що суттєво обмежує масштабованість алгоритму на великі системи. Практично це призводить до труднощів навчання вже при $N > 5–7$ агентів. Крім того, модель MADDPG, як правило, навчається для фіксованої кількості агентів і не забезпечує узагальнення на системи іншої розмірності без додаткового перенавчання.

У роботі Табіш Рашид та співавторів [63] запропоновано алгоритм QMIX, у якому реалізовано підхід монотонної факторизації спільної функції цінності. Зокрема, глобальна Q-функція $Q_{tot}(s, a)$подається як композиція індивідуальних функцій цінності агентів $Q_i(o_i, a_i)$подається через спеціалізовану змішувальну нейромережу, що задовольняє умову монотонності.

Факторизація спільної функції цінності в алгоритмі QMIX формалізується як подання глобальної Q-функції $Q_{tot}(s, a)$ у вигляді композиції індивідуальних функцій агентів [64]:

$$Q_{tot}(s, a) = f(Q_1(o_1, a_1), \dots Q_N(o_N, a_N)) \quad (1.10)$$

Така архітектура забезпечує узгодженість між централізованим навчанням і децентралізованим виконанням, дозволяючи кожному агенту приймати рішення на основі локальних спостережень без втрати глобальної координації у кооперативних середовищах.

Водночас накладене обмеження монотонності звужує клас функцій, які може апроксимувати модель, оскільки виключає можливість представлення довільних залежностей між індивідуальними та спільною функціями цінності. Це, у свою чергу, може призводити до зниження ефективності алгоритму в задачах зі складною структурою взаємодій або наявністю конкурентних інтересів між агентами [64]. Додатковим обмеженням є залежність архітектури

змішувальної мережі від фіксованої кількості агентів, що ускладнює масштабування підходу в умовах змінної або великої розмірності мультиагентної системи.

У роботі Ю. Сії та співавторів [65] запропоновано алгоритм MAPPO, який є адаптацією методу (Proximal Policy Optimization) до мультиагентного середовища. У межах даного підходу реалізується парадигма централізованого навчання з децентралізованим виконанням: для кожного агента навчається централізований критик ($V_i(s)$), що використовує глобальний стан системи, тоді як стратегія агента ($\pi_i(a_i. o_i)$) базується виключно на локальних спостереженнях.

Порівняно з такими підходами, як MADDPG та QMIX, алгоритм MAPPO характеризується підвищеною стабільністю процесу навчання та здатністю досягати вищих показників ефективності у кооперативних задачах завдяки використанню обмеження кроку оновлення політики та більш стійкої процедури оптимізації.

Водночас, у контексті задач управління роєм безпілотних літальних апаратів, стандартна реалізація MAPPO має суттєві обмеження. Зокрема, вона не враховує графову структуру взаємодій між агентами, що є важливим для моделювання локальних комунікацій у рої. Крім того, централізований критик, як правило, отримує на вхід конкатенований вектор станів усіх агентів, що призводить до квадратичної залежності обчислювальної складності від кількості агентів $O(N^2)$ і, відповідно, обмежує масштабованість методу у великих мультиагентних системах [66, 67, 68].

Окрім проблем масштабованості та узагальнення, одним із ключових обмежень класичних алгоритмів мультиагентного навчання з підкріпленням є відсутність вбудованих механізмів гарантування безпеки поведінки агентів. Це особливо важливе у задачах управління роєм безпілотних літальних апаратів, де помилки взаємодії між агентами можуть призводити до зіткнень та втрати стійкості формації. У зв’язку з цим у сучасних дослідженнях значна увага приділяється розробленню підходів Safe-MARL, орієнтованих на забезпечення безпечної координації мультиагентних систем.

Так, в роботі Алшекіна М. та співавторів [69] розроблено підхід Safe-MARL (Safe Multi-Agent Reinforcement Learning), що базується на використанні функції Ляпунова для забезпечення формальних гарантій безпеки в мультиагентних системах. Запропонований метод дозволяє аналітично довести інваріантність безпечної множини станів, однак його застосування обмежене необхідністю задання явної аналітичної форми обмежень, що суттєво ускладнює використання в складних динамічних середовищах. Крім того, підхід характеризується низькою масштабованістю та є обчислювально придатним переважно для систем з невеликою кількістю агентів ($N \leq 3$–4). У контексті порівняльного аналізу зазначені результати підтверджують критичну важливість інтеграції явних механізмів забезпечення безпеки в MARL-системах, водночас вказуючи на доцільність використання альтернативних підходів, зокрема формування функції винагороди з урахуванням штрафів за порушення безпечної поведінки (safety reward), як більш практично орієнтованого рішення.

Амес А. [70] запропонував використання функцій бар'єрного контролю (CBF) як формального інструменту забезпечення безпеки при управлінні багатьма агентами. У рамках даного підходу безпечна множина станів визначається як $C = \{x: h(x) \geq 0\}$, а її інваріантність гарантується виконанням умови $\dot{h}(x) + \gamma h(x) \geq 0$. Інтеграція CBF із нейромережевими методами керування забезпечує можливість реалізації жорстких обмежень безпеки. Водночас застосування цього підходу в мультиагентних системах супроводжується суттєвими обмеженнями, зокрема, врахування попарних взаємодій між агентами призводить до квадратичної обчислювальної складності $O(N^2)$, а реалізація у тривимірному просторі для задач керування квадрокоптерами вимагає розв'язання задач неконвексної оптимізації, що ускладнює практичне впровадження.

Підходи, засновані на використанні потенціальних полів відштовхування (репульсивних потенціалів) як складової функції винагороди в MARL, детально розглянуто в роботах [71, 72]. Узагальнення результатів цих досліджень свідчить, що включення штрафного члена, пропорційного частоті міжагентних

зіткнень ($\beta^{*}CR_{inter}$), у функцію винагороди за умови достатньо великого коефіцієнта ваги β ($\beta \geq$ 5-10 *при* $\alpha = 2$) для компонента похибки підтримання формації забезпечує ефективне навчання агентів дотриманню безпечних дистанцій без необхідності застосування формальних гарантій типу CBF. Експериментальні результати, отримані у реалістичних тривимірних симуляційних середовищах, демонструють досягнення рівня міжагентних зіткнень $CR_{inter} < 2\%$ при збереженні високої точності підтримання формації (Δf < 0,2 м.), що підтверджує практичну ефективність даного підходу.

Подальший розвиток мультиагентного навчання з підкріпленням пов’язаний із необхідністю забезпечення масштабованості та ефективної обробки взаємодій між змінною кількістю агентів. У цьому контексті важливим напрямом є використання механізмів уваги, що дозволяють адаптивно агрегувати інформацію про інших агентів.

Ікбал С. та Ша Ф. [20] запропонували механізм уваги для централізованого критика в MARL, який забезпечує можливість обробки змінної кількості агентів. У межах даного підходу критик формує оцінку цінності шляхом обчислення зваженої суми внесків інших агентів із використанням self-attention механізму. Це дозволяє моделі автоматично визначати релевантність взаємодій між агентами без жорсткого задання структури зв’язків.

Зазначений підхід є концептуальним попередником графових методів у MARL, зокрема критиків на основі механізму уваги на графах, що використовуються у даному дослідженні. Принципова відмінність полягає в тому, що у запропонованій роботі взаємодії між агентами визначаються явно через динамічний граф $G(t)$, у якому зв’язки формуються лише між агентами, що перебувають у межах радіусу комунікації (R_{comm}). На відміну від цього, у підході [20] механізм уваги обчислюється для всіх пар агентів, що призводить до квадратичної обчислювальної складності $O(N^2)$, тоді як використання локалізованої графової структури дозволяє знизити складність до $O(N^{*}k)$ де k – середня кількість сусідів агента.

Питання розподілу ресурсів у мережах безпілотних літальних апаратів із використанням методів опуклої оптимізації розглянуто в роботі [73], що підкреслює актуальність поєднання методів оптимізації та навчання з підкріпленням у задачах координації рою.

Филип Христіанос та співавтори [22] дослідили проблему узагальнення MARL-алгоритмів на системи з іншою кількістю агентів. Зокрема, показано, що алгоритм, навчений для конфігурації з $N = 5$ агентів, демонструє суттєве зниження ефективності при збільшенні кількості агентів до $N = 10$ (зниження показника успішності SR з 87% до 54%) у випадку відсутності механізмів обміну інформацією між агентами. Водночас використання підходів із спільними параметрами та механізмами передачі повідомлень, зокрема на основі графових нейронних мереж, дозволяє значно зменшити деградацію продуктивності ($SR = 84\%$ при $N = 10$). Отримані результати є важливим емпіричним обґрунтуванням доцільності застосування GNN-архітектур у задачах управління роями БПЛА.

Аналогічні підходи до підвищення ефективності навчання мультиагентних систем на основі використання спільного досвіду та графових структур також розглядаються у роботі [74], що додатково підтверджує перспективність інтеграції GNN-компонентів у MARL-моделі.

Отже, проведений аналіз сучасних підходів до мультиагентного навчання з підкріпленням свідчить про необхідність поєднання механізмів забезпечення безпеки, масштабованості та урахування топологічної структури взаємодій між агентами, що обґрунтовує доцільність використання графових нейронних мереж у поєднанні з методами MARL для ефективного управління роями безпілотних літальних апаратів.

1.4.4 Методи на основі графових нейронних мереж

Графові нейронні мережі (GNN) є природним інструментом для моделювання системи агентів, топологія взаємодій яких описується графом.

Ключовою операцією є передача повідомлень, тобто кожен вузол-агент агрегує інформацію від сусідів і оновлює своє внутрішнє представлення [75, 76]:

$$\tilde{h}^{(l+1)} = UPDATE(h_i^{(1)}, AGGREGATE(\{m_{ji}: j \in N(i)\})) \quad (1.11)$$

де m_{ji} -- повідомлення від агента j до агента i (MESSAGE($h_j^{(l)}$, $h_i^{(l)}$),

$N(i)$ – множина сусідів агента і у поточному графі G(t).

Graph Attention Network (GAT) [33] запроваджує механізм уваги в процесі агрегування інформації від суміжних вершин графа, що дозволяє адаптивно зважувати внесок кожного сусіда під час оновлення представлення вузла. На відміну від класичних графових нейронних мереж, де внесок сусідів є фіксованим або залежить лише від структури графа, у GAT ваги визначаються динамічно на основі навчуваних параметрів. Зокрема, для кожної пари суміжних вузлів i та j обчислюється коефіцієнт уваги як функція подібності між їхніми векторними представленнями, що інтерпретується як скалярний добуток відповідних query- та key-векторів. Це дозволяє моделі ідентифікувати релевантність інформації, яка надходить від кожного сусіда, та здійснювати селективне агрегування.

Формально це можна визначити як [77]:

$$\alpha_{ij} = softmax\left(e(h_i, h_j)\right), \qquad \tilde{h}_i = \sigma(\sum j \in N(i) * \alpha_{ij} \ldots . Wh_{ij} \quad (1.12)$$

де W – навчувана матриця трансформації,

e – функція оцінки уваги.

Застосування графових нейронних мереж, зокрема GNN та GAN, у задачах координації рою безпілотних літальних апаратів (БПЛА) демонструє низку принципових переваг порівняно з класичними підходами багатоагентного підкріплювального навчання, такими як MADDPG та QMIX [56, 78].

По-перше, у рамках GNN-підходу критик оперує не конкатенованим глобальним вектором станів усіх агентів, а агрегованим графовим представленням, сформованим на основі локальних взаємодій. Це дозволяє

знизити обчислювальну складність із квадратичної $O(N^2)$, характерної для повнозв'язних представлень, до лінійно-обмеженої $O(N*k)$. Такий підхід суттєво підвищує ефективність обробки інформації в умовах масштабованих багатоагентних систем.

По-друге, архітектури GNN є інваріантними до перестановок вузлів графа, що означає відсутність залежності від фіксованого порядку індексації агентів. Завдяки цьому модель не є «прив'язаною» до конкретних ідентифікаторів сусідів і зберігає узагальнювальну здатність при зміні кількості агентів у системі. Це забезпечує природну масштабованість і можливість перенесення навченої політики на системи з іншими значеннями N без необхідності повторного навчання.

По-третє, використання розріджених графових структур, у яких взаємодія обмежується лише k-найближчими сусідами, узгоджується з фізичними обмеженнями реальних систем зв'язку БПЛА, зокрема з радіусом комунікації R_{comm}. Така локалізація взаємодій не лише зменшує обчислювальні витрати, але й підвищує робастність системи до втрати зв'язку та змін топології мережі, що є критично важливим у динамічних середовищах функціонування рою.

Отже, застосування GNN/GAT-архітектур забезпечує ефективне поєднання обчислювальної масштабованості, топологічної гнучкості та відповідності фізичним обмеженням комунікаційної інфраструктури, що робить їх перспективним інструментом для задач координації багатоагентних систем.

У сучасних дослідженнях координації мультиагентних систем, зокрема роїв безпілотних літальних апаратів, значну увагу приділено використанню GNN як ефективного інструменту моделювання локальних взаємодій між агентами. Перевагою такого підходу є здатність враховувати топологію графа взаємодій та забезпечувати масштабованість обчислень за рахунок локального обміну інформацією. Зокрема, у роботі [79] запропоновано розподілений алгоритм керування роєм БПЛА на основі минимальної апроксимації поля, що забезпечує узгоджений рух та уникнення зіткнень. Подальший розвиток цього напряму пов'язаний із інтеграцією графових нейронних мереж, які дозволяють ефективно

моделювати топологію взаємодії агентів та локальні залежності між ними [80]. Так, у роботі [81] запропоновано поєднання підходу середнього поля (mean-field) з механізмами GNN що забезпечує масштабованість та покращення якості координації в умовах обмеженої спостережуваності. Зазначений підхід передбачає використання усередненого представлення станів сусідніх агентів замість повного графового згорткового проходу (GCN-pass), що дозволяє суттєво зменшити обчислювальну складність та затримки при обробці інформації, забезпечуючи масштабованість системи до 10 БПЛА у 2D-просторі. Кожен агент агрегує середнє представлення сусідів замість повного GCN-pass, що знижує затримку обчислень. Проте запропонований метод не тестується у 3D-середовищі з реалістичною динамікою квадрокоптера, не містить явного обмеження безпечних відстаней і вимагає перенавчання при зміні кількості агентів. Дослідження свідчать, що точність формації у 2D при N=10, $\Delta f \approx 0{,}19$ м у 3D ця цифра значно погіршується через нехтування вертикальною динамікою [79, 80].

У роботі Дж. Ксяо. [77] досліджено застосування механізмів уваги в межах графових нейронних мереж для задачі управління роєм безпілотних літальних апаратів чисельністю 10–20 агентів. Запропонований підхід базується на використанні спрощеного механізму обчислення коефіцієнтів уваги, у якому стандартна softmax-нормалізація замінена σ-нелінійною функцією активації. Така модифікація дозволяє суттєво знизити обчислювальні витрати та забезпечити скорочення затримки обробки інформації приблизно на 40% порівняно з класичною архітектурою Graph Attention Network, що є важливим для систем реального часу.

Водночас запропонований підхід характеризується низкою обмежень. Зокрема, у роботі відсутня реалізація парадигми CTDE (Centralized Training with Decentralized Execution), оскільки централізований критик використовується не лише на етапі навчання, але й під час виконання, що зумовлює необхідність постійного доступу до глобальної інформації та підвищує вимоги до міжагентних комунікацій. Крім того, питання масштабованості не отримало

належного висвітлення, оскільки експериментальні дослідження проведено для фіксованої кількості агентів без аналізу узагальнювальної здатності моделі при зміні розміру рою. Також у запропонованому підході відсутні явні механізми забезпечення безпеки, зокрема обмеження мінімальних міжагентних відстаней, що обмежує його практичне застосування у щільних формаціях БПЛА.

Х. Джоу [82] запропонували інтегрований підхід до управління роєм безпілотних літальних апаратів, що поєднує графові нейронні мережі (GNN) із алгоритмом підкріплювального навчання Proximal Policy Optimization (PPO), реалізований у симуляційному середовищі AirSim. У межах експерименту для рою з восьми агентів ($N = 8$) у сценарії зі статичними перешкодами досягнуто показника успішності виконання завдання на рівні $SR = 78\%$. Водночас отримані результати мають низку обмежень: відсутня верифікація масштабованості підходу для інших розмірів рою ($N \neq 8$), не проведено абляційного аналізу внеску GNN-компонента в загальну ефективність моделі, а також зафіксовано підвищений рівень міжагентних зіткнень ($CR_{inter} = 4{,}2\%$), що суттєво перевищує цільове значення (< 2%), необхідне для безпечної координації.

А. Недич та співатори [9] сформували фундаментальні теоретичні засади розподіленої оптимізації в мережевих багатоагентних системах. Зокрема, доведено збіжність розподілених градієнтних алгоритмів за умови зв’язного динамічного графа взаємодій між агентами. Отримані результати мають важливе методологічне значення для обґрунтування збіжності процесів навчання GNN-акторів у середовищах із топологічно змінною структурою комунікацій рою. Слід зазначити, що аналогічні підходи до забезпечення узгодженості поведінки агентів досліджувалися і вітчизняними науковцями, зокрема в контексті розроблення алгоритмів консенсусу для розподіленого управління мережами безпілотних літальних апаратів [19].

На рисунку 1.2. систематизовано класифікацію методів координації групи БПЛА.

Аналіз сучасних досліджень у сфері багатоагентного підкріплювального навчання [24, 83, 84, 85, 86] свідчить про те, що інтеграція графових нейронних

мереж, зокрема GNN, із алгоритмами політик градієнтного типу, такими як MAPPO, формує один із найбільш перспективних напрямів розвитку методів координації багатоагентних систем, зокрема роїв безпілотних літальних апаратів.

Водночас, незважаючи на значний потенціал такого підходу, існуючі реалізації характеризуються низкою суттєвих обмежень.

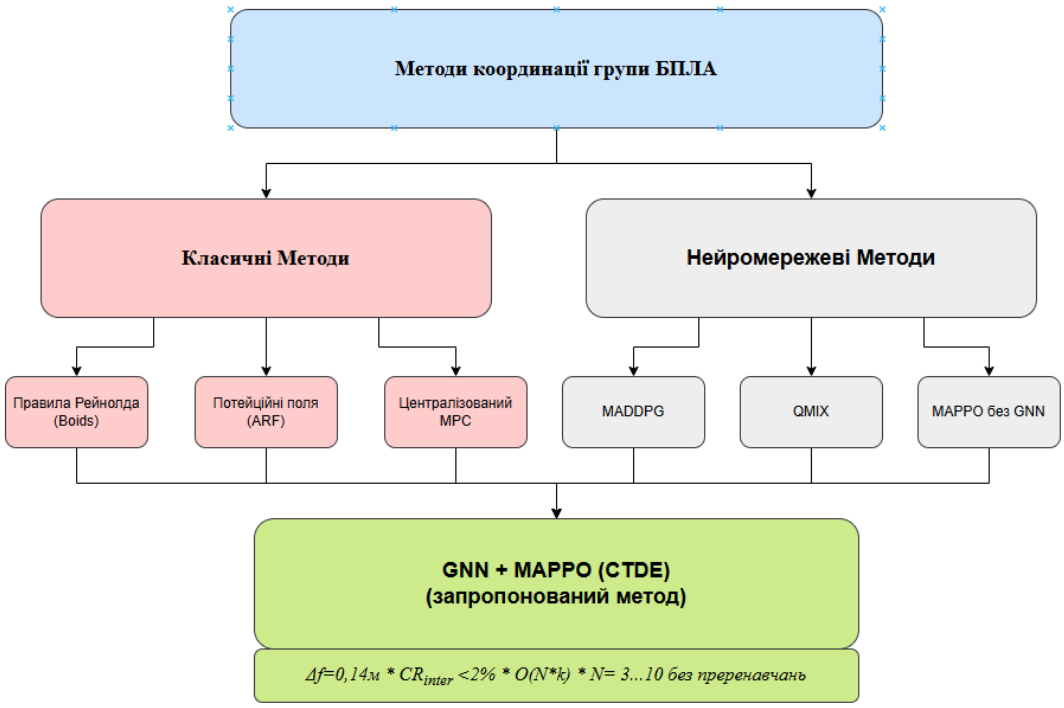

Рис. 1.2. Класифікація методів координації групи БПЛА

По-перше, переважна більшість досліджень обмежується моделюванням у двовимірному просторі, що не враховує складної кінематики та динаміки БПЛА у тривимірному середовищі. Відсутність урахування реалістичних обмежень руху, інерційних характеристик та аеродинамічних ефектів знижує прикладну цінність отриманих результатів.

По-друге, лише обмежена кількість робіт розглядає механізми забезпечення безпечних міжагентних відстаней як явну (explicit) складову архітектури або функції винагороди. У більшості випадків уникнення зіткнень

реалізується опосередковано, що не гарантує дотримання критичних обмежень безпеки в умовах високої щільності агентів або динамічних змін топології.

По-третє, у наявних дослідженнях практично відсутнє системне оцінювання масштабованості моделей. Зокрема, рідко розглядаються сценарії узагальнення, за яких модель, навчена на системі з фіксованою кількістю агентів (наприклад, N=5), тестується на конфігураціях із меншою (N=3) або більшою (N=10) кількістю агентів без додаткового перенавчання. Це питання є суттєво важливим для практичного застосування, оскільки реальні багатоагентні системи характеризуються змінною кількістю учасників та динамічною структурою взаємодій.

Отже, подальші дослідження у напрямі поєднання GNN та MAPPO доцільно орієнтувати на подолання зазначених обмежень, зокрема шляхом переходу до тривимірних середовищ із реалістичною динамікою, інтеграції явних механізмів гарантування безпеки та проведення комплексного аналізу масштабованості моделей. В таблиці 1.2 наведено порівняльний аналіз методів координації групи БПЛА, де CR_{inter} – частота зіткнень між агентами; ЦТ – централізоване тренування; CTDE – Centralized Training, Decentralized Execution; $O(N*k)$ – лінійна складність, k = const (кількість сусідів).

Таблиця 1.2

Порівняльний аналіз методів координації групи БПЛА

Метод	*Δf, м*	CR_{inter}	Масшта-бованість	Децентр. виконання	Адапт. до втрати зв'язку
Reynolds (правила)	0,45-0,80	8-15%	✓ O(N)	✓ Так	✗ Ні
Потенційні поля	0,38-0,65	5-12%	✓ $O(N^2)$	✓ Так	✗ Ні
Централіз. MPC	0,18-0,28	1-3%	✗ $O(N^3)$	✗ Ні	✗ Ні
MADDPG	0,22-0,35	3-8%	✗ фікс. N	✗ ЦТ	Частково
QMIX / MAPPO	0,19-0,30	2-6%	✗ фікс. N	✗ ЦТ	Частково
GNN+MAPPO (запроп.)	0,14-0,20	<2%	✓ $O(N*k)$	✓ CTDE	✓ Так

Джерело сформовано автором за [24, 83, 84, 85, 86]

Проведений аналіз, (див. табл. 1.2), дозволяє зробити наступні висновки, що класичні методи координації рою БПЛА забезпечують обчислювальну простоту та децентралізованість, однак поступаються за показниками безпеки та адаптивності, тоді як централізовані підходи характеризуються високою точністю ціною значної обчислювальної складності та низької масштабованості. Методи багатоагентного підкріплювального навчання (MADDPG, QMIX/MAPPO) дозволяють покращити якість координації, проте залишаються обмеженими фіксованою кількістю агентів і залежністю від централізованого навчання. Натомість запропонований підхід на основі графових нейронних мереж у поєднанні з MAPPO забезпечує найкращий баланс між безпекою, масштабованістю та стійкістю до втрати зв’язку, що підтверджує його перспективність для застосування в динамічних багатоагентних системах.

З метою систематизації розглянутих підходів до координації мультиагентних систем та узагальнення їх ключових характеристик проведемо порівняльний аналіз за основними критеріями, зокрема типом методу, масштабованістю, середовищем апробації та наявними обмеженнями. Це дозволяє не лише виявити сильні та слабкі сторони існуючих рішень, але й обґрунтувати вибір архітектурних рішень. Узагальнені результати порівняння наведено в таблиці 1.3.

Як видно з таблиці 1.3, класичні підходи, зокрема правила Рейнольдса та метод потенційних полів, характеризуються високою обчислювальною ефективністю та природною децентралізованістю, однак не забезпечують необхідного рівня точності підтримання формації та не гарантують безпечної взаємодії між агентами. Методи мультиагентного навчання з підкріпленням, такі як MADDPG, QMIX та MAPPO, демонструють покращення якості координації, проте залишаються обмеженими фіксованою кількістю агентів і характеризуються зростанням обчислювальної складності, що ускладнює їх масштабування.

Таблиця 1.3

Порівняльний аналіз підходів мультиагентного навчання з підкріпленням та класичних методів керування роєм БПЛА

Автори, рік	Метод / Підхід	Кількість агентів	Середовище	Ключові обмеження
1	2	3	4	5
Reynolds (1987)	Правила boids	N > 100	Симулятор	Точність Δf>0,5 м; без гарантій безпеки
Olfati-Saber (2006)	Потенційні поля	$N \leq 50$	Теорія/симул.	Локальні мінімуми; не масштаб. на NN
Lowe et al. (2017)	MADDPG	$N \leq 5$	OpenAI/симул.	Фіксоване N; $O(N^2)$ критик
Rashid et al. (2018)	QMIX	$N \leq 8$	StarCraft II	Фіксоване N; не тестов. на БПЛА
Yu et al. (2022)	MAPPO	$N \leq 10$	SMACv2/симул.	Відсутній механізм безпеки між агентами
Li et al. (2021)	GNN-based MARL	$N \leq 10$	2D симулятор	2D простір; без безпек. відстані
Wang et al. (2022)	GAT + MARL	$N \leq 20$	Симулятор	Відсутнє тест. у 3D; нема CTDE
Zhou et al. (2023)	GNN + PPO	$N \leq 8$	AirSim	Не масштаб. без перенавч.; нема SR
Запропонований підхід	GNN+MAPPO (CTDE)	N = 3-10	Gazebo 3D	Повна верифікація у 5 сценаріях

Джерело: сформовано автором

У свою чергу, підходи на основі графових нейронних мереж забезпечують більш ефективне врахування локальної структури взаємодій між агентами та дозволяють знизити обчислювальну складність за рахунок використання розріджених графів. Разом з тим, аналіз існуючих робіт свідчить про наявність низки невирішених проблем, зокрема обмежену апробацію в тривимірних середовищах, відсутність явних механізмів гарантування безпеки та недостатнє дослідження узагальнювальної здатності моделей при зміні кількості агентів.

Запропонований у даному дослідженні підхід, що поєднує графові нейронні мережі з алгоритмом MAPPO у парадигмі CTDE, спрямований на подолання зазначених обмежень шляхом забезпечення масштабованості, урахування топологічної структури взаємодій та інтеграції механізмів безпеки,

що підтверджує його перспективність для задач координації роїв безпілотних літальних апаратів.

1.5 Постановка задачі дослідження

1.5.1 Формулювання наукової проблеми

Сучасні дослідження у сфері координації мультиагентних систем показують, що підвищення автономності груп БПЛА потребує не лише точного управління рухом та формаціями, а й інтеграції механізмів забезпечення безпеки, масштабованості та обчислювальної ефективності [13; 54, 87]. Традиційні централізовані алгоритми управління обмежені у можливості масштабування та стійкості до втрат окремих агентів, тоді як існуючі децентралізовані підходи не завжди гарантують достатню точність утримання формації та ефективність виконання на ресурсно-обмежених бортових контролерах.

У цьому контексті необхідним є розроблення комплексної децентралізованої моделі управління групою БПЛА, яка б одночасно забезпечувала високі показники точності, безпеки, масштабованості та робастності, з урахуванням обмежених обчислювальних ресурсів бортових систем.

На основі проведеного системного аналізу існуючих методів та підходів наукову проблему можна сформулювати таким чином: необхідно розробити децентралізовану нейромережеву модель управління групою безпілотних літальних апаратів (N = 5 - 20 апаратів), яка забезпечує одночасне виконання наступних вимог:

- висока точність утримання формації де допустиме відхилення від заданої формації $\Delta f \leq 0{,}20$ м;
- гарантії безпеки між агентами (ймовірність зіткнення $CR_{inter} < 2\%$*);*
- масштабованість алгоритму, можливість адаптації до зміни кількості агентів у діапазоні N від 3 до 10 без необхідності повторного навчання;

- обчислювальну ефективність, що дозволяє виконання алгоритму на бортовому ARM-контролері з частотою ≥ 20 Гц;
- робастність системи до втрати зв'язку та відмови окремих агентів.

1.5.2 Формалізація задачі управління

Реалізація децентралізованого управління групою безпілотних літальних апаратів (БПЛА) потребує точного математичного опису взаємодії агентів, їх динаміки та інформаційного обміну в умовах обмеженої спостережуваності. Існуючі методи координації мультиагентних систем не завжди забезпечують одночасне виконання критеріїв точності формації, безпеки, масштабованості та обчислювальної ефективності на ресурсно-обмежених бортових контролерах [13; 54, 87].

У цьому контексті задача управління групою БПЛА формалізується у вигляді кооперативної децентралізований частково спостережуваний процес прийняття рішень Маркова (Decentralized Partially Observable Markov Decision Process (Dec-POMDP)) [60], що дозволяє комплексно враховувати динаміку агентів, стохастичні збурення, обмежену спостережуваність та взаємодію у рої:

$$\mathcal{G} = \langle I, S, \{A_i\}, \{O_i\}, P, \{R_i\}, G(t), \gamma \quad \rangle, \tag{1.12}$$

де I – множина агентів $(\{1, ..., N\})$;

S – глобальний простір станів (положення, швидкість, орієнтація та кутові швидкості кожного агента) $(S \subseteq R^{12N})$;

A_i – простір дій агента і (бажані лінійні та кутова швидкості) $(A_i \subseteq R^4)$;

O_i – простір спостережень $(O_i \subseteq R^{(12+7k)})$, при чому власний стан $s_i \in R^{12}$ та повідомлення від k сусідів $m_{ji} \in R^7$;

P – стохастична функція переходу $(S * A \rightarrow \Delta(S))$;

$G(t)$ – динамічний граф зв'язку рою $(V, E(t))$.

Стан агента визначається:

$$s_i = [p_i, v_i, q_i] \in R^{12}, \tag{1.13}$$

де p_i – положення ($p_i \in R^3$);

v_i – швидкість ($v_i \in R^3$);

q_i – орієнтація (кути Ейлера та кутові швидкості) ($q_i \in R^6$)

Динаміка агента описується рівнянням:

$$\dot{s}_i = f(s_i, u_i) + w_i, \tag{1.14}$$

де w_i – шум (вітрові збурення, сенсорний шум).

Повідомлення агента j до агента i формалізується як:

$$m_{ji} = [p_j - p_i, v_j, form_{ji}] \in R^7, \tag{1.15}$$

де $form_{ji}$ – відносний зсув у формації.

$$form_{ji} = p_j^* - p_i^*. \tag{1.16}$$

Функція винагороди агента I має вигляд:

$$R_i(t) = -\alpha * \Delta f_i - \beta * [d_{min}(i,t) < d_{safe}] - \gamma * [\lambda_{min}(G(t)) < \lambda_{thresh}] + \delta * EE_i, \tag{1.17}$$

де $\alpha, \beta, \gamma, \delta$ – вагові коефіцієнти;

$d_{min}(i,t)$ – мінімальна відстань до найближчого сусіда:

$$d_{min}(i,t) = \min\{j \neq i\} \|p_i - p_j\|$$

$\lambda_{min}(G(t))$ – штраф за втрату зв'язності групи (мінімальне власне значення (алгебраїчна зв'язність) графа G(t));

EE_i – доданок ефективності енергоспоживання.

Задача пошуку оптимальної спільної стратегії агентів:

$$\pi^* = \operatorname{argmax}_\pi E\left[\sum_{t=0}^{T} \gamma^t \sum_{i \in I} R_i(t)\right], \tag{1.18}$$

за умов $\|p_i - p_j\| \geq d_{safe}$ та $\forall i \neq j, \ \forall t$

Таким чином, запропоноване формалізоване представлення Dec-POMDP забезпечує комплексний опис динаміки рою БПЛА, інформаційного обміну та критеріїв оптимізації, що є основою для розробки децентралізованих

нейромережевих алгоритмів управління та їх подальшої експериментальної валідації.

1.5.3 Технічні вимоги до розроблюваної системи

Розробка децентралізованої нейромережевої системи управління групою БПЛА вимагає не лише формалізації математичної моделі взаємодії агентів та пошуку оптимальної політики, а й чіткого визначення технічних вимог до системи. Вони повинні враховувати особливості експлуатації групових роїв, обмеження обчислювальних ресурсів бортових контролерів, безпеку польотів, точність утримання формації, час реакції системи та масштабованість алгоритмів.

На основі проведеного аналізу нормативних документів, сучасних стандартів авіаційної безпеки та практичних вимог до задач картографування, аграрних операцій і SAR-застосувань сформовано комплекс технічних параметрів, що забезпечують ефективну та безпечну роботу системи в реальних умовах. Ці параметри включають точність формації, безпечну відстань між агентами, успішність виконання місії, час збору формації, затримку управління, масштабованість, обсяг пам’яті бортової нейромережевої моделі та інші критично важливі характеристики (див. Табл.1.4)

Значення d_{safe}=2–3 корпуси апарата відповідає практикам організації безпечних групових польотів та нормам авіаційної безпеки. Затримка управління у 200 мс встановлена, як поріг обробки змін у реальному часі, що забезпечує адекватну реакцію системи на критичні навігаційні події, що відбуваються швидше, ніж вони сприймаються візуально.

Цільові значення технічних параметрів обґрунтовані на основі аналізу сучасних нормативних документів та стандартів, зокрема ICAO Doc 10019 [78], EASA Regulation 2019/947 [88], ASTM F3269-17 [89] та ISO 21384-3:2019 [90], що гарантує відповідність системи вимогам безпеки та ефективності при експлуатації в різних умовах.

Таблиця 1.4

Технічні вимоги до децентралізованої нейромережевої системи управління групою БПЛА

Вимога	Параметр	Цільове значення	Обґрунтування
Точність формації	Δf, *м*	$\leq 0{,}20$	Задачі картографування, SAR
Безпека (між агентами)	CR_{inter}, %	≤ 2	Запобігання зіткненням у рої
Успішність місії	SR, %	≥ 90	Аграрні / SAR застосування
Час збору формації	T_{form}, с	≤ 6	Оперативність реагування
Затримка управління	$\tau_{інф}$, мс	≤ 200 (реальний час)	Поріг зорового сприйняття
Масштабованість	N_{max}	≥ 10 без перенавч.	Потреба груп 5-20 БПЛА
Пам'ять моделі (борт)	RAM, МБ	≤ 50	Бортовий ARM-контролер
Безпечна відстань	d_{safe}	2-3 корпуси апарата	Стандарти авіаційної безпеки

Джерело: сформовано автором

Висновки до розділу 1

У першому розділі дисертаційної роботи здійснено комплексний аналіз теоретичних, методичних та прикладних аспектів управління групами безпілотних повітряних суден у контексті сучасних вимог до мультиагентних систем. Встановлено, що перехід від використання одиночних БПЛА до ройових систем є об’єктивною тенденцією розвитку галузі, зумовленою необхідністю підвищення ефективності виконання складних задач у цивільному, інфраструктурному та оборонному секторах. Застосування груп БПЛА дозволяє реалізувати паралельну обробку просторово-розподілених задач, що забезпечує суттєве скорочення часу виконання місій та підвищення продуктивності системи загалом.

Показано, що ключовою науково-технічною проблемою є розроблення ефективних методів координації рою, які одночасно забезпечують узгодженість поведінки агентів, безпечну міжагентну взаємодію та масштабованість при збільшенні кількості БПЛА. Аналіз сучасних досліджень засвідчив, що

традиційні централізовані підходи, незважаючи на здатність забезпечувати глобальну оптимальність, мають обмежену застосовність у реальному часі через високу обчислювальну складність, затримки передачі даних та наявність єдиної точки відмови. Водночас децентралізовані архітектури демонструють вищу стійкість до відмов, адаптивність та відповідність умовам функціонування у динамічних середовищах, однак супроводжуються складністю забезпечення глобальної узгодженості.

Встановлено, що гібридні підходи, зокрема в межах парадигми централізованого навчання з децентралізованим виконанням (CTDE), є найбільш перспективним напрямом розвитку систем управління роями БПЛА. Вони дозволяють поєднати переваги глобального навчання з автономністю прийняття рішень окремими агентами, що забезпечує баланс між ефективністю координації та масштабованістю системи.

У межах розділу обґрунтовано доцільність використання формальної моделі рою у вигляді динамічного графа, що адекватно відображає топологію міжагентних взаємодій, обмеження комунікації та змінність структури зв’язків у часі. Запропонований підхід дозволяє формалізувати ключові характеристики системи, зокрема похибку утримання формації, безпекові обмеження та умови зв’язності, що створює основу для побудови ефективних алгоритмів кооперативного управління.

Визначено систему метрик оцінювання ефективності групового управління, яка включає показники точності (похибка формації), безпечної взаємодії (на основі частоти зіткнень), надійності (на основі частоти успішних місій), динамічних властивостей (за часом формування формації) та масштабованості (збільшення кількості апаратів в групі / рою). Застосування цих метрик забезпечує можливість об’єктивного порівняння різних алгоритмічних підходів та формує єдину методологічну базу для оцінювання їх ефективності.

Аналіз нормативно-правової бази показав, що сучасні міжнародні та національні стандарти [78,88-90] висувають жорсткі вимоги до безпеки, надійності та інтеграції БПЛА у повітряний простір. Зокрема, основними є

вимоги до забезпечення уникнення зіткнень (Detect and Avoid), дотримання мінімальних безпечних відстаней та проведення попередньої оцінки ризиків для групових операцій. Це додатково підкреслює необхідність інтеграції механізмів безпеки безпосередньо в алгоритми управління.

Проведений огляд існуючих методів координації рою БПЛА показав еволюцію підходів від простих евристичних моделей, таких як правила Рейнольдса та метод потенційних полів, до сучасних інтелектуальних методів, зокрема мультиагентного навчання з підкріпленням. Встановлено, що класичні підходи характеризуються високою обчислювальною ефективністю, однак не забезпечують необхідного рівня точності та безпеки. У свою чергу, методи MARL дозволяють формувати складні кооперативні стратегії, проте мають суттєві обмеження щодо масштабованості, узагальнення на змінну кількість агентів та забезпечення безпеки.

Особливу увагу приділено методам на основі графових нейронних мереж, які забезпечують природне врахування топології взаємодій між агентами та дозволяють знизити обчислювальну складність за рахунок локалізації обміну інформацією. Встановлено, що інтеграція GNN із методами мультиагентного навчання з підкріпленням створює передумови для побудови масштабованих, адаптивних та ефективних систем координації роїв БПЛА.

Разом з тим, аналіз сучасних досліджень виявив низку невирішених проблем. Зокрема, існуючі підходи не забезпечують одночасного досягнення високої точності утримання формації ($\Delta f \leq 0{,}2$ м), низького рівня міжагентних зіткнень ($CR_{inter} < 2\%$) та масштабованості без необхідності перенавчання при зміні кількості агентів. Крім того, більшість моделей не враховують повною мірою динамічну змінність топології зв’язку та обмеження реальних каналів комунікації.

Таким чином, проведений аналіз обґрунтовує доцільність розроблення нових підходів до управління групами БПЛА на основі інтеграції графових нейронних мереж із методами мультиагентного навчання з підкріпленням у рамках парадигми CTDE. Такий підхід дозволяє забезпечити узгодженість

поведінки агентів, гарантії безпеки, масштабованість та адаптивність до динамічних умов середовища.

РОЗДІЛ 2 ТЕОРЕТИКО-МАТЕМАТИЧНІ МОДЕЛІ ТА МЕТОДИ ДЕЦЕНТРАЛІЗОВАНОГО УПРАВЛІННЯ РОЄМ БПЛА

2.1 Математична модель багатоагентної системи БПЛА як динамічного графа

Побудова строгої математичної моделі це етап, що передує проєктуванню будь-якого алгоритму управління. На відміну від задачі керування одиночним БПЛА, ройова система характеризується додатковими рівнями складності, які мають як структурний, так і динамічний характер. По-перше, топологія взаємодій між агентами не є фіксованою, а змінюється в часі залежно від їх просторового розташування, обмежень комунікаційного радіуса, умов видимості та впливу зовнішнього середовища. По-друге, стан системи має колективну природу, оскільки динаміка кожного окремого агента визначається не лише його власним станом і керуючими діями, але й станами сусідніх агентів, з якими він безпосередньо взаємодіє. Це призводить до виникнення складної взаємозалежної динаміки, в якій локальні взаємодії формують глобальну поведінку рою [32, 33].

У зв’язку з цим адекватним математичним апаратом для опису таких систем є теорія динамічних графів, яка дозволяє формалізувати як структуру зв’язків, так і процеси обміну інформацією між агентами. Представлення рою у вигляді графа, вершини якого відповідають окремим БПЛА, а ребра – каналам взаємодії між цими БПЛА, дає змогу описати змінну топологію системи як функцію часу та врахувати обмеження локальної комунікації. Такий підхід забезпечує можливість переходу від індивідуального опису агентів до системного рівня аналізу, де ключову роль відіграють характеристики зв’язності, щільності графа, а також властивості локальних сусудів кожного агента.

Крім того, графове подання створює формальну основу для опису процесів координації, консенсусу, утримання формації та уникнення зіткнень у термінах локальних правил взаємодії. Це, у свою чергу, важливо для синтезу

децентралізованих і масштабованих алгоритмів управління, зокрема на основі методів мультиагентного навчання та графових нейронних мереж, де обчислення керуючих дій безпосередньо залежить від структури міжагентних зв’язків.

2.1.1 Кінематика агента та вектор стану

Кожен агент рою $i \in I = \{1, \ldots, N\}$є квадрокоптером зі стандартною X-конфігурацією чотирьох роторів. Стан агента *i* у момент часу *t* описується вектором $s_i(t) \in R^{12}$, що містить усі необхідні кінематичні [25, 58] характеристики для управління та комунікації. Детальний опис систем навігації, керування та бортового автопілота наведено у відповідних літературних джерелах [91]. Враховуючи всі необхідні кінематичні характеристики для управління та обміну інформацією, стан агента iii у момент часу ttt задається наступним вектором:

$$s_i(t) = [p_{\text{i}}^{\text{T}}, v_{\text{i}}^{\text{T}}, \eta_{\text{i}}^{\text{T}}, \omega_{\text{i}}^{\text{T}}]^{\text{T}} \in R^{12}, \tag{2.1}$$

де $p_{\text{i}} = [x_{\text{i}}, y_{\text{i}}, z]^{\text{T}} \in R^3$ – вектор положення у інерціальній системі координат EFF (Earth-Fixed Frame) [м];

$v_i = [vx_{\text{i}}, vy_{\text{i}}, vz_{\text{i}}]^{\text{T}} \in R^3$ – вектор лінійної швидкості у EFF [м/с];

$\eta_{\text{i}} = [\varphi_{\text{i}}, \theta_{\text{i}}, \psi_{\text{i}}]^{\text{T}} \in R^3$ – вектор кутів Ейлера (крен, тангаж, рискання) за ZYX-конвенцією [рад];

$\omega_{\text{i}} = [p_{\text{i}}, q_{\text{i}}, r_{\text{i}}]^{\text{T}} \in R^3$ – вектор кутових швидкостей у зв'язаній системі координат BFF (Body-Fixed Frame) [рад/с].

Нелінійна динаміка агента описується рівняннями Ньютона-Ейлера:

$$\dot{s}_i(t) = f(s_i(t), u_i(t)) + w_i(t) \tag{2.2}$$

де $u_{\text{i}}(t) = [v^{\text{x}}des, v^{\text{y}}des, v^{\text{z}}des, \psi des]^{\text{T}} \in R^4$ – бажані лінійна швидкість та кутова швидкість рискання (вихід policy Actor-мережі);

$w_{\text{i}}(t) \sim N(0, \Sigma w)$ – шум стану (вітрові збурення, шум сенсорів);

$f\cdot(s_i(t))$ – функція переходу, що реалізована у симуляторі ArduPilot SITL з моделлю DJI F450.

Методологічні засади верифікації алгоритмів керування безпілотними літальними апаратами у симуляційних середовищах, а також підходи до імітаційного моделювання роїв розглянуто у відповідних літературних джерелах [59, 92, 93, 94]. Подані підходи та результати досліджень дозволяють систематизувати компоненти стану агента та визначити їх нормалізацію для подальшої подачі на вхід графової нейронної мереж. Детальний склад вектора стану агента та процедура його нормалізації для подачі на вхід нейронної мережі наведені у таблиці 2.1.

Таблиця 2.1

Компоненти вектора стану агента s_i та їх нормалізація для входу GNN

Складова стану	Позначення	Розмір. вектора	Діапазон нормаліз.	Джерело даних	Частота оновл.	Особливості
Положення	$p_i \in R^3$	3	±500 м → [−1,1]	GNSS/RTK	20 Гц	EFF, WGS-84
Лінійна швидкість	$v_i \in R^3$	3	±15 м/с → [−1,1]	GNSS + EKF	20 Гц	EFF
Кути Ейлера (ZYX)	$\eta_i \in R^3$	3	±π рад → [−1,1]	IMU + EKF	100 Гц	φ,θ,ψ
Кутові швидкості	$\omega_i \in R^3$	3	±10 рад/с → [−1,1]	IMU гіроскоп	100 Гц	BFF
Цільовий зсув δ_i	$\delta_i \in R^3$	3	±10 м → [−1,1]	місія	1 Гц	формац. офсет від центру
Відхилення формації	$\Delta f_i \in R^3$	3	±5 м → [−1,1]	обчислення	20 Гц	$p_i - p^*_i$

Джерело: складено автором за [59, 92, 93, 94]

Отже, проведений аналіз узагальнює ключові компоненти стану агента та їх нормалізацію, забезпечуючи основу для подальшого моделювання та застосування в алгоритмах керування роїв БПЛА.

2.1.2 Динамічна графова модель рою

Система з N агентів формалізується у вигляді динамічного неорієнтованого графа $G(t) = (V, E(t))$, де $V = \{1, ..., N\}$ є фіксованою множиною вершин, що відповідають агентам рою, $E(t) \subseteq V \times V$ є змінною в часі множиною ребер-зв'язків. Така постановка дозволяє адекватно враховувати зміну топології взаємодій між агентами в процесі руху рою. Ребро $(i, j) \in E(t)$ між агентами i та j існує тоді і лише тоді, коли евклідова відстань між ними не перевищує радіус комунікації R_{comm}:

$$(i, j) \in E(t) \quad \leftrightarrow \quad \|p_i(t) - p_j(t)\|^2 \leq R_{comm} \tag{2.3}$$

Таким чином, структура графа безпосередньо визначається просторовим розташуванням агентів та їхніми комунікаційними можливостями. Рис. 2.1 ілюструє приклад динамічного графа зв’язку рою $G(t) = (V, E(t))$ при зміні топології.

Для кількісного опису топологічних властивостей графа вводяться матриця суміжності. Зв'язність $G(t)$ характеризується матрицею суміжності $A(t) \in \{0,1\}^{N \times N}$ та матрицею ступенів $D(t) = diag(d_1, \ldots, dN)$, де $d_i(t) = |N(i, t)|$ – кількість сусідів агента i.

На основі цих матриць визначається графа Лапласіан:

$$L(t) = D(t) - A(t), \tag{2.4}$$

Який є одним із основних інструментів аналізу зв’язності та колективної динаміки рою.

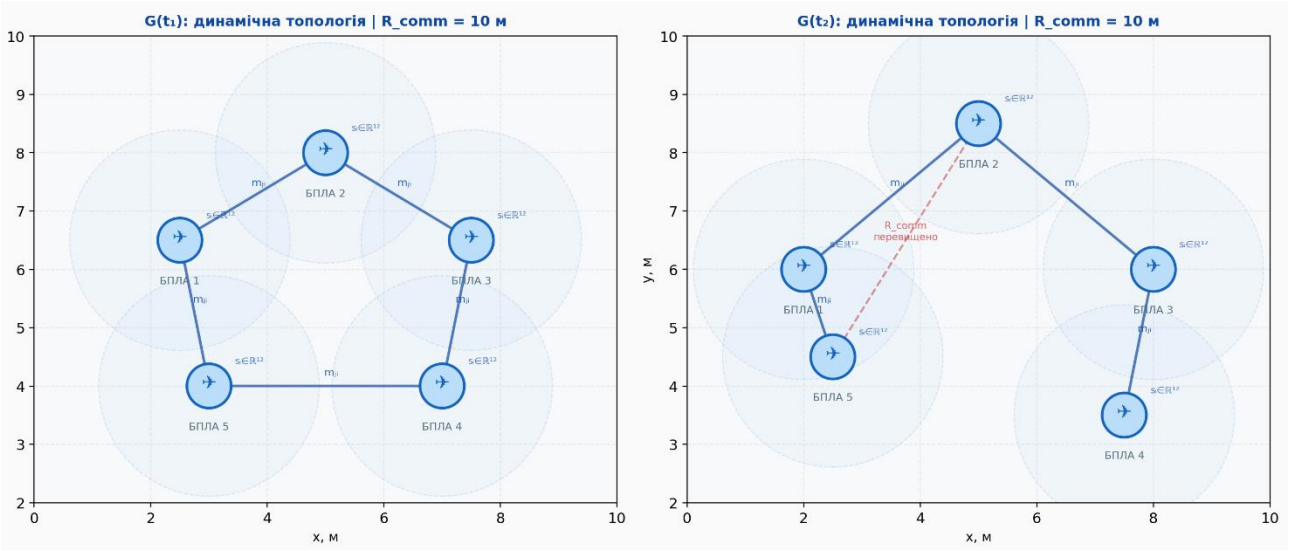

Рис. 2.1. Динамічний граф зв'язку рою $G(t) = (V, E(t))$ при зміні топології

Зокрема, алгебраїчна зв’язність рою характеризується другим найменшим власним значенням $\lambda_2(L(t)) > 0$ – так зване число Фідлера, яке дорівнює нулю тоді і лише тоді, коли граф незв’язний. Таким чином, для забезпечення координованого управління роєм як єдиною системою необхідно виконання умови $\lambda_2(L(t)) > \lambda_{thresh}$ [32, 33].

Для кожного агента і вводиться множина k-найближчих сусідів у радіусі R_{comm}:

$$N(i,t) = \{ j \in V : (i,j) \in E(t) \}, \ |\mathcal{N}(i,t)| = d_i(t) \leq k_{max} \quad (2.4)$$

З метою локалізації взаємодій та зменшення обчислювальної складності для кожного агента iii вводиться множина його найближчих сусідів у межах радіуса R_{comm}.

Параметр $k_{max} = 6$ обрано з урахуванням практичних аспектів функціонування роїв безпілотних літальних апаратів, оскільки у більшості реальних сценаріїв просторові конфігурації квадрокоптерів характеризуються наявністю не більше ніж $k \leq 6$ найближчих сусідів за умов їх відносно рівномірного розподілу у тривимірному просторі.

Водночас розрідженість матриці суміжності $A(t)$ за умови $k \ll N$ є визначальним фактором забезпечення обчислювальної ефективності графових

нейронних мереж, оскільки дозволяє досягти лінійної складності $O(N*k)$ порівняно з квадратичною складністю $O(N^2)$, характерною для повнозв'язних підходів [75, 76].

Таким чином, введене обмеження на кількість сусідів забезпечує баланс між адекватністю моделювання локальних взаємодій у рої та ефективністю реалізації алгоритмів керування на основі GNN.

2.1.3 Конфігурація формації та метрики відхилення

На основі введеної графової моделі взаємодії агентів розглянемо задачу формування та підтримання заданої конфігурації рою. Математичні моделі та методи управління роєм БПЛА досліджуються в роботах багатьох вчених, зокрема у роботі [95].

Цільова конфігурація формації задається набором зсувів $\{\delta_i\}_i \in I$ відносно центра рою $p_c(t)$:

$$p*_{\text{i}}(t) = p_c(t) + \delta_{\text{i}}, \quad \text{де. } p_c(t) = \left(1/N\right) * \Sigma_{\text{i}} \in I\ p_{\text{i}}(t)$$

Таким чином, положення кожного агента у формації визначається його відносним зсувом відносно центру мас рою. Вектор $\delta_i \in R^3$ є фіксованим параметром місії і визначає геометричну форму формації (горизонтальна решітка, вертикальна стінка, куля тощо).

При цьому, в умовах децентралізованого управління, кожен агент знає лише власний δ_i та отримує $\delta_j - \delta_i$ від сусідів, як частину повідомлення m_{ji}. Для оцінювання якості утримання формації вводиться метрика відхилення агента i від його цільового положення:

$$\Delta f_i(t) = \|p_i(t) - p*_i(t)\|^2 \quad [м].$$

На основі індивідуальних відхилень визначаються агреговані показники якості формації, зокрема середня похибка по групі та максимальне відхилення:

$$\Delta f(t) = \left(\frac{1}{N}\right) * \Sigma_{\text{i}} \in I\ \Delta f_{\text{i}}(t),\ \ \Delta f_{max}(t) = max\{i \in I\}\ \Delta f_{\text{i}}(t) \qquad (2.5).$$

Зазначені метрики дозволяють комплексно оцінювати ступінь відповідності поточної конфігурації рою заданій формації.

Поряд із точністю утримання формації важливим аспектом є забезпечення безпечної взаємодії між агентами. Безпекова умова між агентами для всіх пар $i \neq j$ та всіх $t \in [0, T]$:

$$\|p_i(t) - p_j(t)\|^2 \geq \mathrm{d}_{\mathrm{safe}} \tag{2.6}$$

де d_{safe} задається у одиницях корпусів апарата (2-3 корпуси, що відповідає 0,6-0,9 м для апаратів класу DJI F450 з діагоналлю ~ 0,45 м) відповідно до рекомендацій авіаційної практики групових польотів [36, 37].

Таким чином, сформульовані умови забезпечують як точність підтримання формації, так і безпеку функціонування рою в умовах децентралізованого управління.

2.2 Протокол передачі повідомлень між агентами

Комунікаційна складова архітектури визначає, яку саме інформацію агенти обмінюються між собою та як вона кодується для подачі у GNN. Цей вибір безпосередньо впливає на обчислювальну складність і здатність алгоритму узагальнюватися на нові конфігурації рою [78, 88, 89]. Нейромережеві методи обробки інформації в авіаційних системах, включаючи нечіткі нейронні мережі, що досліджено у роботах [96, 97]. З огляду на зазначене, подальший розгляд зосереджено на формалізації протоколу передачі повідомлень між агентами рою.

2.2.1 Структура повідомлення між агентами

У межах розроблюваного протоколу взаємодії, кожен агент j передає агенту i повідомлення $m_{ji} \in R^7$, що містить мінімально необхідний набір даних для координації та безпеки [98, 99]:

$$m_{\mathrm{ji}} = [p_{\mathrm{j}} - p_{\mathrm{i}};\ \ v_{\mathrm{j}};\ \ \|\delta_{\mathrm{j}} - \delta_{\mathrm{i}}\|^2] \in R^7 \tag{2.7}$$

де $p_j - p_i \in R^3$ − відносне положення сусіда у EFF;

$v_j \in R^3$ − абсолютна швидкість сусіда;

$\|\delta_j - \delta_i\|_2 \in R^1$ −норма відносного зсуву у цільовій формації.

Обрана структура повідомлення забезпечує компактне представлення просторово-кінематичної інформації, достатньої для прийняття рішень у децентралізованому режимі керування.

Структуру повідомлення та семантику кожного поля наведено у таблиці 2.2.

Таблиця 2.2

Структура повідомлення $m_{ji} \in R^7$ між агентами рою

Позиції у m_{ji}	Позначення	Розмір ність	Нормалізація	Семантика	Роль при агрег.
1-3	$p_j - p_i \in R^3$	3	$\pm R_{comm}$ м → $[-1,1]$	Відносне положення сусіда j щодо агента i	Розмір графа
4-6	$v_j \in \mathbb{R}^3$	3	±15 м/с → [−1,1]	Абсолютна швидкість сусіда j	Прогноз руху
7	$form_{ji} \in \mathbb{R}$	1	±10 м → [−1,1]	Скалярна норма відносного зсуву у формації $\delta_j - \delta_i$	Координація

Джерело: складено автором

Принципово важливою особливістю запропонованого підходу є те, що повідомлення m_{ji} не включає глобальних координат p_j, а лише відносне положення $p_j - p_i$.

Це означає, що кожен агент потребує лише відносної інформації про сусідів, яка може бути отримана за допомогою бортових сенсорів (ультразвукові датчики, LiDAR, UWB тощо). Водночас використання глобальних навігаційних систем (GNSS) обмежується оцінкою власного стану s_i, що підвищує надійність функціонування рою в умовах деградації або втрати сигналів глобального позиціонування.

2.2.2 Агрегація повідомлень та інваріантність до перестановок

Визначальною властивістю будь-якого алгоритму управління роєм є інваріантність до перестановки агентів, а результат агрегації не повинен залежати від того, в якому порядку нумеруються сусіди. Забезпечення цієї властивості гарантує, що один і той самий алгоритм може коректно функціонувати для будь-якого складу рою та будь-якої топології графа без необхідності перенавчання або зміни ваг мережі.

Загальна операція агрегації на l-му шарі GNN:

$$\tilde{h}_i^{(l+1)} = UPDATE(h_i^{(l)},\ AGGREGATE(\{MESSAGE(h_j^{(l)}, h_i^{(l)}, e_{ji}) : j \in \mathcal{N}(i)\})) \quad (2.8)$$

де MESSAGE – функція формування повідомлення (реалізована як спільна MLP);

AGGREGATE – інваріантна до перестановок функція агрегації (у GAT - зважена сума з attention-вагами α_{ij});

UPDATE – функція оновлення стану агента (MLP + skip-з'єднанням).

Стандартні операції агрегації, такі як сумування, середнє або максимальне значення, за визначенням є інваріантними до перестановок. Механізм уваги (GAT) забезпечує збереження структури при перестановках, що є більш сильною властивістю порівняно зі звичайними інваріантними функціями, і дозволяє диференційовано впливати на внесок кожного сусіда під час агрегації. Отже, реалізація збереження структури при перестановках визначає основу для побудови ефективних протоколів обміну повідомленнями та подальшого керування формацією рою.

2.3 Формальна постановка задачі

Задача координованого управління роєм БПЛА формалізується у рамках кооперативного Decentralized Partially Observable Markov Decision Process (Dec-

POMDP) – найбільш загальної формальної моделі для децентралізованих мультиагентних систем з частковою спостережуваністю [60, 61].

Формально система задається кортежем:

$$\mathcal{G} = \langle I, S, \{\mathcal{A}_i\}_i \in I, \{\mathcal{O}_i\}_i \in I, P, \{R_i\}_i \in I, G(t), \gamma \rangle \quad (2.9)$$

де $I = \{1, \ldots, N\}$ –множина агентів;

$S \subseteq R\text{^}(12N)$ – глобальний простір станів системи;

$\mathcal{A}_i \subseteq R^4$ – простір дій агента i;

$\mathcal{O}_i$ – простір спостережень агента i;

$P : S \times A \rightarrow \Delta(S)$ – стохастична функція переходу;

$G(t) = (V, E(t))$ – динамічний граф зв'язку;

$\gamma \in [0,1]$ – коефіцієнт дисконтування.

Така формальна модель дозволяє кількісно описати взаємодію агентів, обмеження на доступну інформацію та стохастичність середовища, що є необхідною умовою для побудови ефективних алгоритмів координації та оптимізації поведінки рою.

2.3.1 Простір спостережень та дій

Для формалізації Dec-POMDP необхідно чітко визначити, яку інформацію агент має доступною та які дії він може виконувати. Спостереження агента i - локальна інформація, доступна без глобального зв'язку і визначається як:

$$o_i(t) = [s_i(t); \delta_i;\ \{m_{ji}(t)\}\{j \in \mathcal{N}(i)\}] \in R\text{^}(18 + 7 \cdot d_i)] \quad (2.10)$$

де $s_i(t) \in R^{12}$ – власний стан;

$\delta_i \in R^3$ – цільовий зсув у формації;

$\delta_i - p_c \cdot est \in R^3$ – оцінений зсув до цільової позиції (p_c оцінюється через mid-point сусідів);

$\{m_{ji}\}\{j \in \mathcal{N}(i)\}$ – повідомлення від $d_i \leq k_{max}$ сусідів.

Розмірність спостереження є змінною (залежить від $d_i(t)$), що вимагає архітектури, здатної обробляти граф довільного розміру -саме це забезпечує GNN. Простір дій задається як:

$$a_i(t) = [v^x des, v^y des, v^z des, \psi des]^T \in [-v_{max}, v_{max}]^3 \times [-\dot{\psi}_{max}, \dot{\psi}_{max}] \quad (2.11)$$

де v_{max} = 5 м/с,

$\dot{\psi}_{max}$ = $\pi/4$ рад/с.

Дії подаються як установленні точки до бортового контролера PX4 через MAVLink; низькорівнева стабілізація орієнтації та висоти виконується PX4-автопілотом.

2.3.2 Функція переходу та спільна функція винагороди

Функція переходу $P(s'|s, a)$ є стохастичною внаслідок вітрових збурень, шуму сенсорів та неточності моделі. У симуляційному середовищі Gazebo вона реалізована через ArduPilot SITL з моделлю квадрокоптера Iris. Спільна функція винагороди агента i:

$$R_i(t) = \alpha \cdot r^{form}{}_i(t) + \beta \cdot r^{safe}{}_i(t) - \gamma \cdot r_{rep,i}(t) + \delta \cdot r^{conn}(t) - \zeta \cdot r^{eff}{}_i(t) \quad (2.12)$$

де кожен доданок відповідає одному аспекту поведінки рою.

Детальний опис та вагові коефіцієнти компонент функції винагороди наведено у таблиці 2.3.

Вибір вагових коефіцієнтів обґрунтований такими міркуваннями. Пріоритет α = 1,0 (точність формації) є основним навчальним сигналом. Штраф β = 8,0 (порушення d_{safe}) у 8 разів більший за максимальний корисний сигнал $|r^{form}| \leq 1{,}0$, що гарантує, що уникнення зіткнень є безумовним пріоритетом для агента.

Таблиця 2.3

Компоненти функції винагороди $R_i(t)$ та їх вагові коефіцієнти

Ваговий коеф.	Формула компоненти	Змістовна назва	Діапазон значень	Роль у навчанні
$\alpha = 1,0$	$r^{form} = -\|p_i - p^*_i\|$	Похибка формації	Лінійна → [−5, 0]	Головний сигнал навчання; заохочує утримання формації $\Delta f_i \to 0$
$\beta = 8,0$	$r^{safe} = -[d_min < d_{safe}]$	Порушення дистанції	Бінарна штрафна {0, −8}	Основний механізм безпеки; при порушенні $d_{safe} > 0$ - одноразовий штраф β
$\gamma = 2,0$	$r_rep(d_{ij}) = \eta/d_{ij}^2$	Репульсивний потенціал	[0, +∞) → clip до 4	М'яке відштовхування до порушення d_{safe}; активний у зоні $[d_{safe}, 2\cdot d_{safe}]$
$\delta = 0,3$	$r^{conn} = \lambda_min(G(t))$	Алгебраїчна зв'язність	[0, N] → нормалізована	Штраф за фрагментацію рою; λ_min → 0 означає відокремлення підгрупи
$\zeta = 0,1$	$r^{eff} = -\|u_i\|^2$	Витрати управління	Від'ємна квадратична	Регуляризація; перешкоджає надмірним маневрам

Репульсивний потенціал $\gamma \cdot r_{rep}$ активується у «м'якій» зоні $[d_{safe}, 2\cdot d_{safe}]$ і формує градієнт відштовхування до настання жорсткого порушення. Бонус зв'язності $\delta \cdot r^{conn}$ штрафує фрагментацію рою -ситуацію, коли підгрупа агентів втрачає зв'язок з рештою, проілюстровано на рис. 2.2.

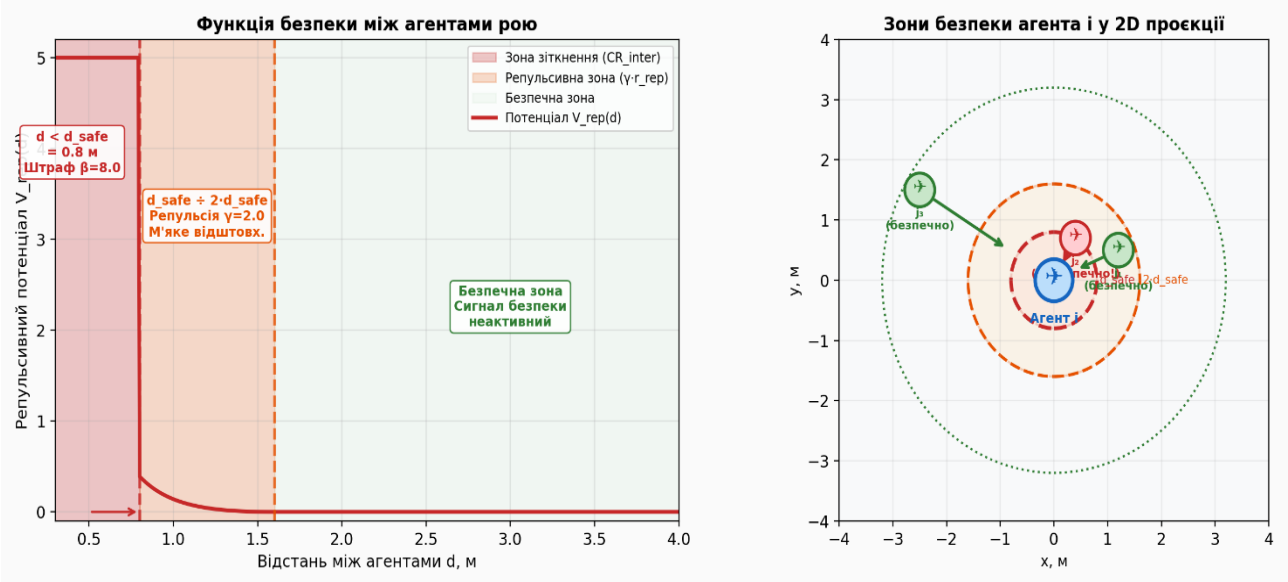

Рис. 2.2 . Функція безпеки між агентами рою та зони потенційних полів відштовхування

Отже, спільна функція винагороди інтегрує важливі аспекти поведінки рою (підтримку формації, безпеку, утримання зв’язності та економію ресурсів управління) та слугує основним сигналом для навчання агентів у рамках Dec-POMDP, забезпечуючи збалансовану та безпечну координацію рою.

2.4 Архітектура мережі уваги на графах

Парадигма централізованого навчання з децентралізованим виконанням є концептуальним вирішенням фундаментального протиріччя між необхідністю глобальної оптимізації на етапі навчання та вимогою автономного функціонування агентів в умовах реального розгортання системи [22, 23].

На етапі навчання застосовується централізована мережа критика, що має доступ до повного глобального стану рою. Така архітектура дозволяє уникнути проблеми нестаціонарності середовища, яка виникає при одночасному незалежному навчанні кількох агентів – коли кожен агент сприймає зміни у поведінці інших як непередбачувані флуктуації середовища, що унеможливлює стабільну збіжність навчального процесу. Структурну організацію парадигми централізованого навчання з децентралізованим виконанням наведено на рисунку 2.3.

На етапі виконання місії кожен агент функціонує виключно на основі власної мережі стратегії керування та локальних спостережень o_i, не потребуючи глобального інформаційного обміну. Це забезпечує повну децентралізацію виконавчого рівня системи та її незалежність від наявності централізованої комунікаційної інфраструктури в умовах реального польоту.

Таким чином можна зробити висновок, що парадигма централізованого навчання з децентралізованим виконанням забезпечує стабільність навчального процесу та надійну поведінку агентів у реальному часі, що є критично важливим для практичної реалізації роїв БПЛА.

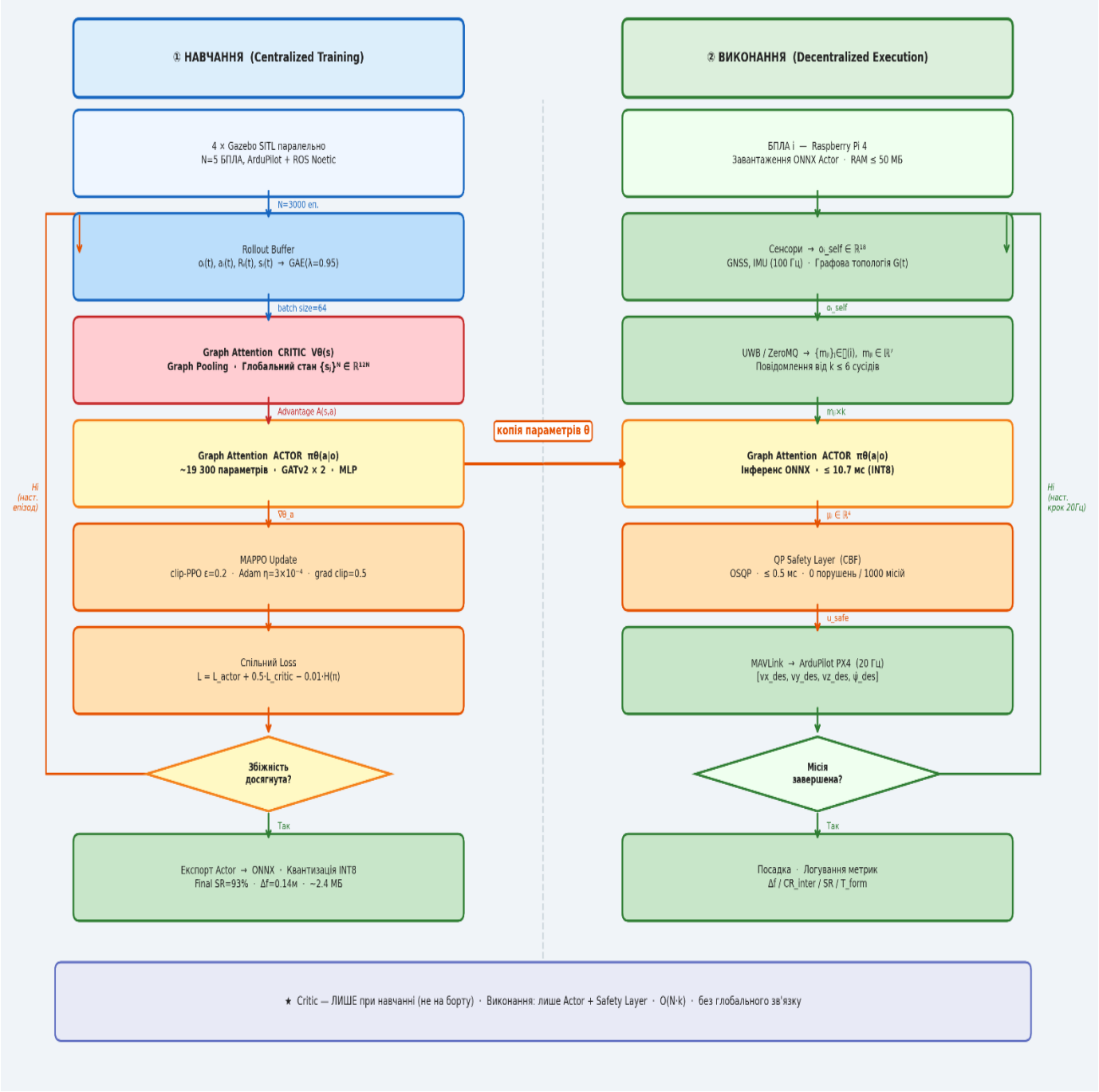

Рис. 2.3. Архітектура системи управління роєм БПЛА за парадигмою CTDE

2.4.1 Актор уваги на графі (децентралізоване виконання)

Актор уваги на графі $\pi_i(a_i \mid o_i)$ є policy-мережею агента i, що визначає розподіл дій на основі локального спостереження. Архітектура Actor будується за принципом двоступеневого message passing з механізмом уваги GAT:

Крок 1. Кодування власного стану: $e_i = Linear(s_i) \in R^{32}$ – проєкція вектора стану агента у приховане представлення.

Крок 2. Кодування повідомлень сусідів: $\tilde{e}_{ji} = MLP_{msg}(m_{ji}) \in R^{32}$ для кожного $j \in \mathcal{N}(i)$.

Крок 3. GAT-агрегація з механізмом уваги (L = 2 шари, K = 4 голови):

$$e_{ij} = LeakyReLU(a^T[We_i \parallel W\tilde{e}_{ji}]), \quad \alpha_{ij} = exp(e_{ij}) / \Sigma_k \in \mathcal{N}(i)\, exp(e_{ik})$$

$$\hbar_i^{(l)} = \parallel \{k = 1\}^{K} \sigma(\Sigma_j \in \mathcal{N}(i)\, \alpha_{ij}^{(k)} \cdot W_k^{(l)} \tilde{e}_{ji})$$

де $\parallel$ – конкатенація K голів уваги: W, a – навчувані параметри; σ = ELU.

Вагова функція уваги α_{ij} навчається автоматично, агент «фокусується» на найбільш релевантних сусідах залежно від контексту (близькість до порушення безпеки, відхилення формації).

Рис. 2.4 демонструє механізм динамічного зважування сусідів агентом через GAT.

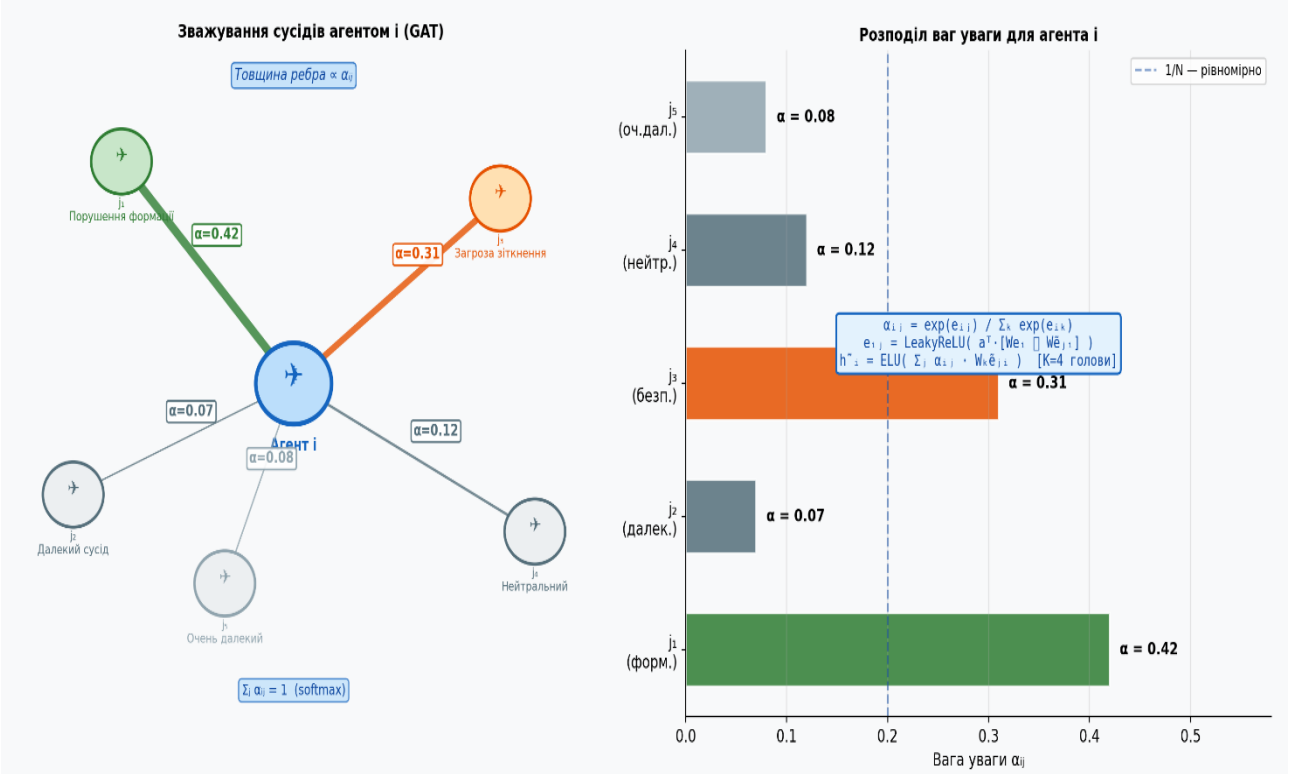

Рис. 2.4. Механізм уваги GAT (динамічне зважування сусідів агентом)

Крок 4. Об'єднання: $z_i = [s_i \,\|\, \tilde{h}_i^{(2)}] \in R^{76}$ – конкатенація власного стану та агрегованого графового представлення.

Крок 5. MLP-декодер: $\mu_i = MLP_out(z_i) \in R^4$, $log\ \sigma_i \in R^4$ – параметри Гауссівської стратегії $\pi(a_i \mid o_i) = \mathcal{N}(\mu_i, diag(\sigma_i^2))$.

Детальний опис кожного шару Актор-мережі наведено у таблиці 2.4.

Таблиця 2.4

Пошарова архітектура Актор уваги на графі

Шар	Тензор / позначення	Розмір.	Параметр	Операція	Коментар
Вхід	$o_i \in \mathbb{R}^{12}$	12	-	Linear norm.	Власний стан s_i (12) + мета δ_i (3) + Δf_i (3) ← стандартизовані
Повідомл.	$m_{ji} \in \mathbb{R}^7 \times k$	$7 \times k$	-	Shared linear	Повідомлення k сусідів; $k \leq 6$ (k-nearest у R_{comm})
GAT шар 1	$\tilde{h}_i^{(1)} \in \mathbb{R}^{64}$	64	4 голови	Multi-head att.	Агрегація повідомлень сусідів; ELU-активація; concat 4 голів → 64
GAT шар 2	$\tilde{h}_i^{(2)} \in \mathbb{R}^{64}$	64	4 голови	Multi-head att.	Другий крок message passing; skip-connection з 1-го шару
Concat	$[s_i; \tilde{h}_i^{(2)}] \in \mathbb{R}^{76}$	76	-	Concatenate	З'єднання власного стану та агрегованого представлення графа
MLP-1	$\in \mathbb{R}^{128}$	128	-	ReLU + BN	Спільне представлення стану-графу
MLP-2	$\in \mathbb{R}^{64}$	64	-	ReLU	Проміжний шар
Вихід Actor	$\mu_i, log\ \sigma_i \in \mathbb{R}^4$	4 + 4	-	Linear / tanh	Параметри Гауссівської policy: бажані [vx,vy,vz,ψ̇]; σ -дисперсія

На рисунку 2.5 наведено графічну ілюстрацію пошарової архітектури Актор уваги на графі, що демонструє взаємодію кодування стану агента, агрегації повідомлень сусідів та формування децентралізованої політики.

Загальна кількість навчуваних параметрів Актор:

- MLP_msg (7→32): 7*32+32 = 256;
- GAT-шар 1 (4 голови, 32→16 кожна, concat→64): 4*(32+32+1) = 260;

- GAT-шар 2 (64→64): 4*(64+64+1) = 516;
- MLP_out (76→128→64→4): 76*128 + 128*64 + 64*4 = 18240.

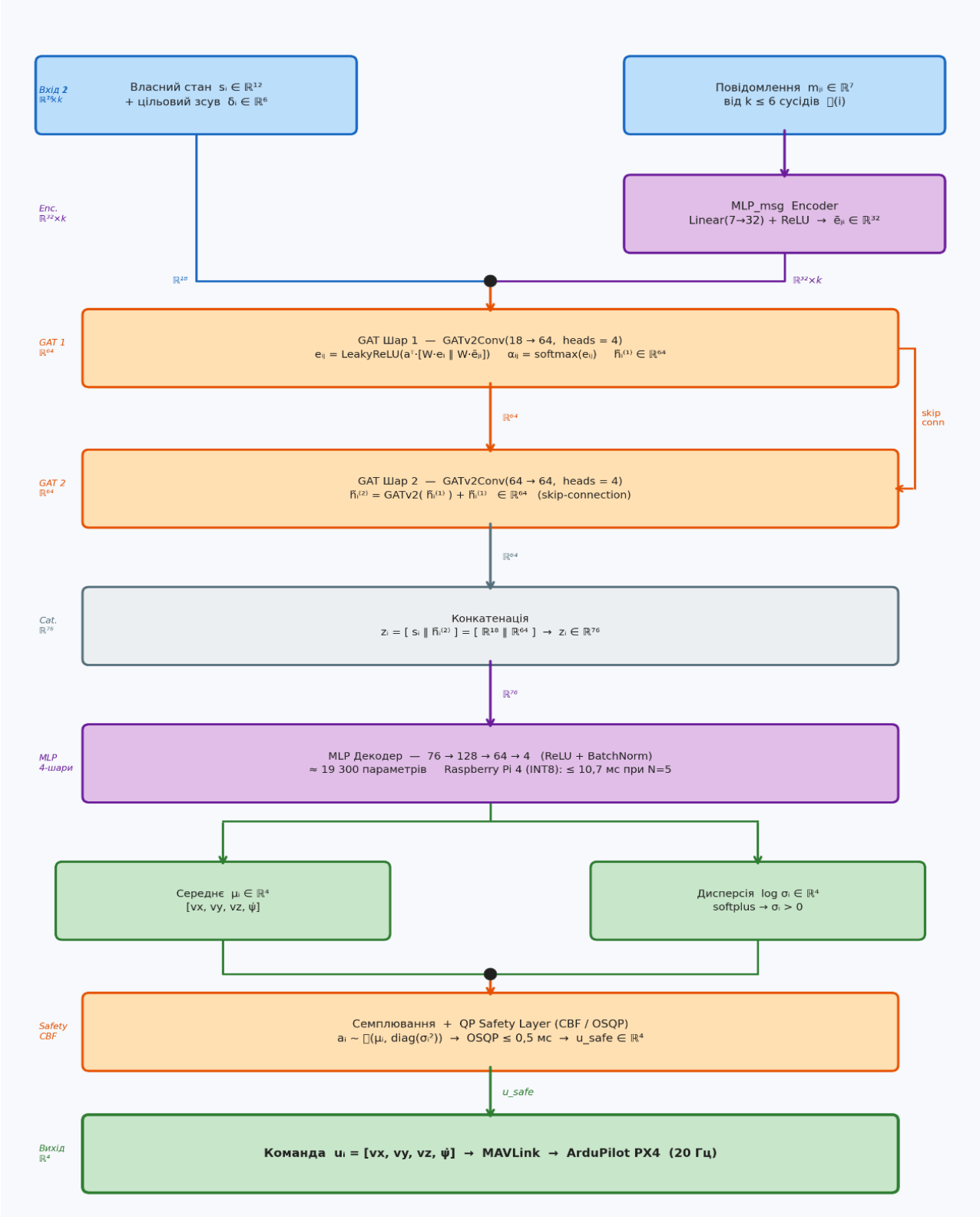

Рис. 2.5. Пошарова архітектура Актор уваги на графі

Загальна кількість навчуваних параметрів Актор становить приблизно 19300, що у 3,4 рази менше, ніж у попередній версії TactNet (~ 65000), при принципово вищій масштабованості. Це підтверджує ефективність GAT-архітектури в контексті децентралізованого навчання та виконання.

2.4.2 Критик уваги на графі (централізоване навчання)

Критик уваги на графі $V_i(s)$ оцінює цінність глобального стану рою для агента і під час навчання. На відміну від Актор, критик не використовується на етапі виконання і не обмежений часовими вимогами, що дозволяє застосовувати більш обчислювально складні операції. Ключова відмінність полягає в тому, що Критик отримує повний стан усіх N агентів, який агрегується через механізм уваги по вершинах графа:

$$V_{input} = GraphPooling(\{\tilde{h}_j / j \in I\}) = \Sigma_j \in I\ \beta_j * W_o h_j \quad (2.12)$$

$$\beta_j = exp(h_j^T * q) / \Sigma_k \in I\ exp(h_k^T * q), \quad (2.13)$$

де $q \in R^{64}$ − навчуваний запитальний вектор (query);

β_j − скалярна вага агента j у глобальному агрегуванні.

Такий підхід дозволяє критику обробляти довільне N (не фіксоване) завдяки інваріантний до перестановок агрегуванню, і забезпечує складність $O(N)$ для forward pass замість $O(N^2)$ у конкатенаційних підходах (MADDPG, QMIX). Вихід критик повертає скалярну оцінку $V_i \in R^1$ що відображає очікуваний кумулятивний дохід агента i, оцінений на основі поточного глобального стану рою і використовується для навчання політики актору.

2.5 Механізм гарантування безпеки між агентами

Безпека між агентами рою забезпечується на двох рівнях: «м'якому» (через функцію винагороди під час навчання) та «жорсткому» (через детерміністичний

шар перевірки при виконанні). Двоступеневий підхід є необхідним, оскільки навчена нейромережа не може давати формальних гарантій безпеки у статистичному сенсі – лише навчений рефлекс уникнення [85, 86, 100, 101].

2.5.1 Репульсивні потенційні поля у функції винагороди

Репульсивна складова функції винагороди $r^{safe}_i + r_{repi}$ реалізує дворівневий потенціал Леннарда-Джонса у просторі міжагентних відстаней. «м'яка» зона ($d_{safe} \leq d_{ij} \leq 2 \cdot d_{safe}$):

$$r_{repi}(t) = \gamma * \Sigma_j \in \mathcal{N}(i)\, max(0, \eta/d_{ij}{}^2 - \eta/(2 * dsafe)^2), \quad (2.14)$$

де $\eta = 0{,}1$ − масштабний коефіцієнт $d_{ij} = \|p_i - p_j\|^2$.

Потенціал активується поступово при наближенні і зростає обернено-квадратично, що створює плавний градієнт відштовхування. У межах «жорсткої » зони ($d_{ij} < d_{safe}$) запроводжуються штраф $r^{safe}_i = -\beta =$ - 8,0 – одноразовий великий від'ємний доход, що унеможливлює ігнорування порушення з боку агента. Така постановка винагороди виключає можливість ігнорування порушення мінімально допустимої дистанції з боку агента та стимулює дотримання обмежень безпеки.

2.5.2 Явний шар перевірки безпечної відстані при виконанні

На рівні виконання між виходом актор-мережі та відправкою команди бортовому контролеру розміщений детерміністичний рівень безпеки. Він реалізує проєкцію бажаної дії на безпечний підпростір:

$$u_{safe} = argmin\{u\} \|u - \mu_i\|^2,\ s.t.\ \|p_i + u * \Delta t - p_j\| \geq d_{safe}\ \ \forall j \in \mathcal{N}(i) \quad (2.15)$$

У випадку, коли запропонована мережею актора керуюча дія μ_i вже задовольняє всі накладені обмеження безпеки, вона передається до виконання

без жодних модифікацій. У разі порушення обмежень формулюється задача квадратичного програмування з кількістю обмежень k, що дорівнює кількості сусідніх агентів у межах $k = |\mathcal{N}(i)| \leq 6$, та чотирма змінними оптимізації. Задача розв'язується розв'язувачем OSQP із застосуванням теплого старту за час не більше 0,5 мс.

Таким чином, рівень безпеки вносить додаткову обчислювальну затримку, що не перевищує 0,5 мс, та надає наступну гарантію: незалежно від того, яку дію пропонує мережа актора, дія, що фактично виконується, буде безпечною відносно поточного положення сусідніх агентів.

Разом з тим необхідно зазначити суттєве застереження щодо меж застосовності запропонованого підходу. Рівень безпеки не є строгим гарантом у сенсі функцій бар'єрної безпеки у випадках значних відносних швидкостей між агентами – зокрема, при зустрічному русі зі швидкістю понад 8 м/с. Це обумовлено тим, що обмеження накладаються виключно на поточний крок керування, а не на прогнозовану майбутню траєкторію руху агентів. Повноцінне забезпечення безпечної поведінки рою досягається насамперед завдяки навченій стратегії уникнення зближень, сформованій у процесі навчання з підкріпленням, тоді як рівень безпеки виконує роль додаткового захисного механізму для граничних та нештатних ситуацій.

2.6 Алгоритм навчання мультиагентній адаптації алгоритму PPO

Навчання агентів здійснюється за алгоритмом Multi-Agent Proximal Policy Optimization (MAPPO) – мультиагентній адаптації алгоритму PPO [66, 67]. Вибір даного підходу є методично обґрунтованим і зумовлений сукупністю його теоретичних та практичних переваг у задачах координації багатьох агентів [66, 67].

По-перше, використання clip-обмеження забезпечує контрольоване оновлення стратегії та сприяє її квазі-монотонному покращенню без різких деградацій.

По-друге, належність до навчання на поточній стратегії підвищує стабільність у порівнянні з не стратегічним підходом, зокрема підходом мультиагентного алгоритму навчання з підкріпленням (Multi-Agent Deep Deterministic Policy Gradient), за рахунок зменшення впливу застарілого досвіду в умовах нестаціонарності.

По-третє, спільна параметризація актор-мережі зменшує обчислювальну складність і забезпечує перенесення знань між агентами при змінній кількості *N*.

По-четверте, централізований критик з механізмом уваги на графі є інваріантним до кількості агентів, що забезпечує масштабованість і ефективне моделювання їх взаємодій.

2.6.1 Математичні основи PPO

Ціль оптимізації актору у PPO -− максимізація сурогатної функції з clip-обмеженням [66]:

$$L^{\{CLIP\}(\theta)} = E_t[min(r_t(\theta) * \hat{A}_t,\ clip(r_t(\theta), 1-\varepsilon, 1+\varepsilon) * \hat{A}_t)], \quad (2.16)$$

де $r_t(\theta) = \pi\theta(a_t|o_t) / \pi_old(a_t|o_t)$ − відношення ймовірностей нової та старої policy;

$\hat{A}_t$ − оцінка переваги (advantage);

$\varepsilon = 0{,}2$ − коефіцієнт clip.

Застосування clip-обмеження дозволяє уникнути надмірних змін стратегії під час оновлення параметрів, що є ключовим чинником підвищення стабільності градієнтних методів навчання стратегії.

Оцінка переваги здійснюється за методом узагальненной оцінки переваги GAE (Generalized Advantage Estimation, λ = 0,95):

$$\hat{A}_t = \sum_{i=0} \delta_{t+i}^{\{T-t-1\}(\gamma\lambda)^i},\ \text{де}\ \delta_t = R_t + \gamma * V(s_{t+1}) - V(s_t) \quad (2.17)$$

Повна ціль навчання актор з ентропійним бонусом:

$$L(\theta) = L^{\{CLIP\}(\theta)} - \mathrm{c}_1 * L^{\{VF\}(\theta)} + c_2 * H[\pi\theta(\cdot\,|o_\mathrm{t})], \qquad (2.17)$$

де $L^{\{VF\}} = E_t[\left(V\theta(s_\mathrm{t}) - V_t^{target}\right)^2]$ – функція втрат критика;

H – ентропія policy;

$c_1 = 0{,}5$; $c_2 = 0{,}01$.

Таким чином, наведена математична постановка забезпечує збалансоване поєднання стабільності оновлення стратегії, точності оцінки функції цінності та достатнього рівня дослідження середовища, що є особливо важливим для ефективного навчання агентів у складних мультиагентних системах.

2.6.2 Алгоритм MAPPO для рою агентів

У MAPPO всі N агентів використовують спільні ваги актор-мережі. Це означає, що при зміні N від 5 до 10 Актор-мережа не змінюється – змінюється лише кількість агентів, що одночасно використовують її. Критик також є спільним, але обробляє глобальний стан рою. Алгоритм навчання складається з трьох циклічних фаз:

Фаза 1. Збір досвіду**:** паралельне розгортання N агентів у 4 середовищах Gazebo ($4N = 20$ агентів одночасно). Протягом T *rollout* = 500 кроків кожен агент зберігає кортеж (o_{it}, a_{it}, R_{it}, o_{it+1}). Разом буфер містить $4 \cdot N \cdot \mathrm{T} = 10000$ записів.

Фаза 2. Обчислення переваг**:** за допомогою централізованого Критик обчислюється $V(s_t)$ для кожного глобального стану, потім GAE-оцінка $\hat{\mathrm{A}}_\mathrm{t}$ для кожного агента.

Фаза 3. Оновлення мережі**:** 10 епох стохастичного градієнтного спуску по міні-батчах розміром 256. Gradient clipping (норма $\leq 0{,}5$) забезпечує стабільність.

Отже, циклічна організація процесу навчання, що поєднує збір досвіду, оцінювання переваг та оновлення параметрів мереж, дозволяє формувати

узгоджену колективну поведінку агентів і забезпечує високу ефективність MAPPO в задачах керування мультиагентними системами.

2.6.3 Гіперпараметри навчання

Повний перелік гіперпараметрів алгоритму MAPPO та обґрунтування вибору їх числових значень наведено у таблиці 2.5. Значення гіперпараметрів визначено за результатами систематичного пошуку по сітці, що охоплював п'ять рівнів варіювання для кожного з чотирьох ключових параметрів: швидкості навчання актора η_a, швидкості навчання критика η_c, параметра відсікання довірчої області ε та коефіцієнта дисконтування γ. Основним критерієм відбору оптимальної конфігурації слугував показник успішності виконання місій *SR* при конфігурації рою з п'яти агентів після 1000 навчальних епізодів.

Таблиця 2.5

Гіперпараметри алгоритму MAPPO та обґрунтування

Гіперпараметр	Значення	Обґрунтування / коментар
1	2	3
Кількість агентів N (навчання)	5	Базовий сценарій; тестування на N = {3, 5, 10}
Кількість епізодів	3 000	Збіжність досягається за ~2 000 епізодів
Довжина епізоду T, кроків	500	25 с при Δt = 0,05 с; охоплює типові місії
Паралельних середовищ Gazebo	4	Асинхронний збір досвіду (vecenv)
Розмір буфера rollout	2 000	Кроків до оновлення мережі
Кількість epochs SGD	10	Повторне використання одного буфера
Розмір mini-batch	256	Рядків досвіду за одне оновлення
PPO clip ratio ε	0,2	Стандарт Schulman et al. (2017)
Коефіцієнт дисконтування γ	0,99	Довгостроковий горизонт
GAE λ (advantage estimation)	0,95	Баланс зміщення-дисперсії
Швидкість навчання Actor η_a	3×10^{-4}	Adam optimizer; cosine annealing
Швидкість навчання Критик η_c	1×10^{-3}	Критик навчається швидше Actor
Коефіцієнт ентропійного бонусу	0,01	Заохочення дослідження
Gradient clipping (норма)	0,5	Стабілізація навчання
Кількість голів GAT	4	*Multi-head attention; concat → ×4*
Розмірність прихованого шару	64	Actor + Критик; баланс ємності/латентності

2.7 Модель збурень та рандомізація домену

Забезпечення ефективного перенесення синтезованої стратегії керування із середовища комп'ютерного моделювання в реальні умови експлуатації є однією з ключових проблем у галузі навчання з підкріпленням для систем із множинними агентами. Вирішення цієї проблеми потребує явного включення до структури симуляційного середовища репрезентативних моделей основних джерел невизначеності. Стосовно груп безпілотних літальних апаратів такими джерелами виступають: параметрична невизначеність динамічних моделей окремих апаратів, атмосферні збурення вітрового характеру, інструментальні похибки бортових сенсорів, а також нестаціонарність умов міжагентного інформаційного обміну.

Атмосферні вітрові збурення моделюються відповідно до стандарту MIL-HDBK-1797B із застосуванням моделі Дрейдена, яка передбачає суперпозицію двох складових:

- детермінованої постійної компоненти $\bar{v}W$ з діапазоном швидкостей до 8 м/с при рівномірно розподіленому напрямку,
- та стохастичної турбулентної компоненти $\delta vW(t)$, що формується фільтром першого порядку з інтенсивністю σu до 2 м/с і просторовим масштабом турбулентності Lu = 150 м.

Похибки супутникового визначення місцеположення описуються нормально розподіленим шумом із середньоквадратичним відхиленням σp у діапазоні 1,5-3,0 м. Окремо відтворюється сценарій часткової недоступності навігаційного сигналу: з імовірністю p_{loss} = 5-20% вимірювання місцеположення вважається відсутнім протягом інтервалу від 1 до 10 секунд, що відповідає умовам сценарію *C5* – втрати навігаційного зв'язку.

Похибки інерціального вимірювального блоку враховуються через адитивну модель, що включає детерміноване зміщення нульового сигналу за прискоренням bak та випадкову шумову складову з відхиленням σa = 0,05 м/с².

Випадкова варіація параметрів моделей здійснюється на початку кожного навчального епізоду шляхом незалежної вибірки фізичних параметрів для кожного агента відповідно до розподілів, наведених у таблиці 2.6. Запропонований підхід дозволяє відтворити реалістичний сценарій функціонування неоднорідного рою, у складі якого можуть одночасно перебувати апарати, що суттєво відрізняються за аеродинамічними та масо-інерційними характеристиками – що є типовою ситуацією при комплексному застосуванні безпілотних систем різних класів і конфігурацій.

Таблиця 2.6

Параметри рандомізація домену для агентів рою

Параметр	Номінал	Відхилення	Розподіл	Обґрунтування
Маса апарата m, кг	1,35	±20%	Uniform	Різні корисні навантаження
Моменти інерції Jx, Jy, кг·м²	0,0082	±25%	Uniform	Механічна розбіжність
Аеродин. гальмування d, Н·с/м	0,055	±30%	Log-Uniform	Мінливий аеродин. опір
Коефіцієнт тяги kT	$1{,}28 \cdot 10^{-8}$	±15%	Uniform	Деградація двигунів
Шум акселерометра σa, м/с²	0,05	*[0,5; 3,0]*	Log-Uniform	Різні класи IMU
Швидкість вітру $\bar{v}W$, м/с	0	[0; 8]	Uniform	Від штилю до 8 м/с
Затримка команд, мс	0	[0; 50]	Uniform	Мережеві затримки
Радіус зв'язку R_{comm}, м	10	*[0,7; 1,3]*	Uniform	Різні RF-умови
Початковий розкид агентів, м	0	[0; 3]	Uniform	Стохастичний старт

Коливання R_{comm} у рандомізації домену (рядок 8 таблиці 2.6) є специфічним для задачі рою та відсутнє в одноагентних сценаріях. Зміна R_{comm} змінює топологію графа $G(t)$ – при меншому R_{comm} граф стає більш розрідженим, що є більш складним сценарієм для GNN-актор, оскільки агент отримує менше повідомлень від сусідів.

2.8 Методологія верифікації та метрики якості

Оцінювання якості функціонування автономних систем пов’язане з аналізом ризиків прийняття рішень у динамічному середовищі. Нейромережеві моделі оцінювання ризику в авіаційних системах дозволяють формалізувати вплив часових та логічних параметрів дій на безпечність місії

Верифікація розробленої системи керування проводиться відповідно до суворої експериментальної методології, що гарантує статистичну значущість отриманих результатів та їх відтворюваність у незалежних дослідженнях. Кожен тестовий сценарій відтворюється у 30 статистично незалежних запусках із різними початковими значеннями генератора псевдовипадкових чисел. Узагальнені результати подаються у формі медіани з міжквартильним розмахом, що забезпечує робастність оцінок до наявності викидів, а також у вигляді 95% довірчого інтервалу, побудованого методом бутстреп-перевибірки з кількістю ітерацій $B = 1000$.

Статистична значущість відмінностей між досліджуваними архітектурами перевіряється за допомогою непараметричного рангового критерію Вілкоксона при рівні значущості $\alpha = 0{,}01$. Вибір зазначеного критерію замість класичного критерію Стьюдента обумовлений характером розподілів ключових метрик якості керування – зокрема показників CR_{inter} та T_{form}, – які демонструють виражену правосторонню асиметрію та важкі хвости розподілу. За таких умов застосування параметричних критеріїв є некоректним з точки зору математичної статистики.

Тестова матриця включає п'ять сценаріїв послідовно зростаючої складності:

- С1 – рій із п'яти апаратів у статичному безперешкодному середовищі з формацією типу «решітка»;
- С2 – рій із п'яти апаратів в умовах динамічних перешкод;
- С3 – рій із п'яти апаратів зі зміною конфігурації формації в ході виконання місії;

- С4 – розширений рій із десяти апаратів для оцінювання масштабованості архітектури;
- С5 – рій із п'яти апаратів в умовах деградації каналу зв'язку до рівня $R_{comm}/2$ протягом 10 секунд для перевірки робастності системи до часткової втрати міжагентної комунікації.

Цільові значення верифікаційних метрик для кожного із зазначених сценаріїв наведено в таблиці 2.7.

Таблиця 2.7

Метрики верифікації та цільові значення для запропонованої архітектури

Вимога	Метрика	Цільове значення	Метод вимірювання	Обґрунтування
1	2	3	4	5
Точність формації	Δf, *м*	≤ 0,20	Середнє по 30 запусках	Картографування: потрібно < 20 см
Безпека між агент.	CR_{inter}, *%*	< 2	Відсоток часових кроків	ASTM F3269 safety envelope
Успішність місій	*SR, %*	≥ 90	Відсоток завершених місій	SAR / аграрні застосування
Час збору формації	T_{form}, *с*	≤ 6	Від старту до Δf < 0,3 м	Оперативність реагування
Затримка виконання	*τ_інф, мс*	≤ 200	Actor forward pass, ARM	Поріг реального часу
Масштабованість	*EI = SR/N*	≥ 0,09	*N* = {3, 5, 10} без перенавч.	*EI_N*=5 / *EI_N*=10 ≥ 0,9
Зв'язність рою	*λ_min(G(t))*	> 0	Усі кроки місії	Відсутність фрагментації
Пам'ять Actor	*RAM_actor, МБ*	≤ 50	PyTorch model size	Бортовий ARM-контролер

Методологія абляційного дослідження передбачає послідовне вимкнення компонентів архітектури для виявлення внеску кожного з них:

- без GAT (замінено на mean-pooling);
- без критик уваги на графі (критик отримує конкатенований вектор);
- без Safety Reward ($\beta = \gamma = 0$ у функції винагороди);
- без Safety Layer (жорсткий шар вимкнено);

- без рандомізація домену (навчання на фіксованих параметрах).

Порівняльний аналіз ефективності запропонованої архітектури здійснюється відносно трьох базових методів: алгоритму Рейнольдса, архітектури MADDPG навченої при $N = 5$, та централізованого регулятора на основі прогнозуючого керування при $N = 5$. Усі базові методи тестуються на ідентичних тестових сценаріях з однаковими початковими значеннями генератора псевдовипадкових чисел, що забезпечує коректність порівняльного аналізу.

Висновки до розділу 2

У другому розділі сформовано повну теоретичну основу децентралізованого нейромережевого керування роєм безпілотних літальних апаратів. Розроблені теоретичні положення охоплюють математичну модель багатоагентної системи у вигляді динамічного графа, формальну постановку оптимізаційної задачі в рамках кооперативного децентралізованого частково спостережуваного марківського процесу прийняття рішень, архітектуру графової нейронної мережі з парадигмою централізованого навчання та децентралізованого виконання, дворівневий механізм забезпечення міжагентної безпеки, а також методологію статистичної верифікації отриманих результатів.

Систему з N агентів формалізовано у вигляді динамічного графа $G(t) = (V,\ E(t))$ із правилом формування комунікаційних зв'язків за радіусом Rcomm. Вектор стану агента $s_i \in R^{12}$ включає повний набір кінематичних характеристик: положення, лінійну швидкість, кути орієнтації та кутові швидкості у відповідних системах координат. Протокол міжагентних повідомлень $m_{ji} \in R^7$ містить мінімально достатній набір координаційних даних – відносне положення сусіднього агента, його швидкість та зсув у формації – і реалізується виключно засобами бортових сенсорів без залежності від глобальної супутникової навігаційної системи.

Функція індивідуальної винагороди $R_i(t)$ включає п'ять складових із обґрунтованими ваговими коефіцієнтами: співвідношення $\beta/\alpha = 8$ закріплює безумовний пріоритет вимог безпечної взаємодії над точністю утримання формації, тоді як складові репульсивного потенціалу та бонусу зв'язності забезпечують м'яке регулювання заданої структури формації рою безпосередньо в процесі навчання.

Запропонована архітектура актора з механізмом уваги на графі містить близько 19300 навчуваних параметрів і реалізована на основі двох шарів графових мереж уваги з $K = 4$ головами та розмірністю прихованого представлення 64. Архітектура Критик доповнена операцією глобального агрегування по графу. Властивість інваріантності до перестановок агентів забезпечує узагальнення на довільну кількість агентів N без перенавчання. Двоступеневий механізм міжагентної безпеки, що поєднує репульсивний потенціал у вигляді функції винагороди із розрахунком безпечної взаємодії на основі задачі квадратичного програмування з часом розв'язання не більше 0,5 мс, є оригінальним вдосконаленням порівняно з наявними підходами на основі графових нейронних мереж.

Алгоритм навчання MAPPO зі спільними параметрами специфіковано з повним набором гіперпараметрів. Модель збурень, що охоплює атмосферну турбулентність, інструментальні шуми інерціального вимірювального блоку та супутникової навігаційної системи, а також випадкову варіацію дев'яти параметрів фізичної моделі, забезпечує реалістичні умови для перенесення навченої стратегії керування в реальне середовище. Методологія верифікації, що передбачає 30 статистично незалежних запусків для кожного з п'яти тестових сценаріїв із застосуванням непараметричного критерію Вілкоксона при рівні значущості $\alpha = 0{,}01$ та оцінюванням за комплексом метрик Δf, CR_{inter}, SR, T_{form} та EI, формує строгу кількісну основу для інтерпретації результатів верифікації у четвертому розділі. Розроблена у цьому розділі теоретична специфікація слугує вичерпним концептуальним базисом для практичної реалізації архітектури, що описується у третьому розділі дисертаційної роботи.

РОЗДІЛ 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ СИСТЕМИ ДЕЦЕНТРАЛІЗОВАНОГО УПРАВЛІННЯ РОЄМ БПЛА

3.1 Загальна архітектура програмного прототипу

Програмний прототип системи децентралізованого керування роєм безпілотних літальних апаратів реалізовано мовою програмування Python 3.9 із залученням бібліотеки глибокого навчання PyTorch 2.0 та спеціалізованої бібліотеки PyTorch Geometric 2.3, орієнтованої на навчання на графових структурах даних. Загальний обсяг кодової бази становить близько 4 000 рядків вихідного коду, організованих за модульною структурою єдиного репозиторію. Вибір бібліотеки PyTorch Geometric обумовлений наявністю оптимізованих операцій розрідженого передавання повідомлень між вершинами графу, підтримкою динамічних графових структур зі змінною кількістю вершин, а також вбудованою інтеграцією з обчислювальною інфраструктурою PyTorch для апаратного прискорення на графічних процесорах.

В основу архітектурного рішення покладено принцип суворого розмежування трьох незалежних рівнів: середовища симуляції, алгоритму навчання та моделей нейронних мереж. Кожен із зазначених рівнів оформлено як самостійний програмний пакет із чітко визначеним інтерфейсом взаємодії. Така організація коду забезпечує можливість незалежної заміни окремих компонентів системи без модифікації суміжних модулів – зокрема, перехід від середовища симуляції Gazebo до AirSim, або заміну алгоритму навчання MAPPO на MADDPG, може бути здійснено без внесення змін до коду нейромережевих моделей.

Залежності програмного середовища зафіксовано у конфігураційному файлі pyproject.toml та відтворюються через середовище conda.

До складу ключових залежностей входять: PyTorch 2.0.1 з підтримкою обчислень на графічному процесорі через CUDA 11.7, бібліотека PyTorch

Таблиця 3.1

Структура кодової бази прототипу

Модуль / пакет	Технологія	Ключові файли	Рядків коду	Призначення
swarm_env/	Gazebo 11 + ROS Noetic	multi_uav_env.py, ros_bridge.py	~580	Multi-UAV симуляційне середовище, N агентів
gnn_model/	PyTorch 2.0 + PyG 2.3	gat_actor.py, gat_critic.py, gnn_utils.py	~640	Актор уваги на графі та Критик уваги на графі
mappo/	PyTorch 2.0	mappo_trainer.py, rollout_buffer.py, ppo_update.py	~720	Алгоритм MAPPO: збір досвіду, GAE, оновлення мережі
safety/	Python + OSQP	inter_safety.py, repulsion.py, qp_filter.py	~380	Міжагентний Safety Layer та репульсивний потенціал
comm/	Python + ZeroMQ	message_protocol.py, graph_builder.py	~290	Протокол m_{ji}, побудова G(t), управління топологією
eval/	Python	metrics.py, ablation.py, scalability.py	~420	Метрики Δf/ CR_{inter} /SR/EI, абляційний аналіз
deploy/	ONNX Runtime 1.16	export_onnx.py, arm_bench.py, sitl_bridge.py	~310	Експорт Actor, бенчмарк на ARM, SITL-інтеграція
tests/	pytest + hypothesis	unit/, integ/, smoke/	~460	Автоматизоване тестування усіх компонентів

Geometric 2.3.1, розв'язувач задач квадратичного програмування OSQP 0.6.3, операційна система для роботів ROS Noetic з програмним інтерфейсом Python 3, імітатор польоту ArduPilot SITL версії 4.4, а також бібліотека асинхронного обміну повідомленнями ZeroMQ 4.3, що реалізує протокол міжагентної передачі повідомлень m_{ji}.

Відтворюваність навчального процесу забезпечується фіксацією початкових значень усіх генераторів псевдовипадкових чисел – стандартної бібліотеки Python, бібліотеки numpy, фреймворку PyTorch та фізичного рушія симулятора Gazebo – через єдиний параметр конфігурації seed = 42.

Таким чином, обрана програмна архітектура забезпечує ефективну реалізацію, тестування та подальше розгортання розробленого прототипу мультиагентної системи.

3.2 Мультиагентне симуляційне середовище Gazebo

Мультиагентне симуляційне середовище забезпечує можливість відтворення динаміки взаємодії агентів та оцінювання ефективності розроблених алгоритмів у контрольованих умовах. Воно дозволяє моделювати складні сценарії координації, враховувати обмеження реального середовища функціонування та проводити відтворювані експерименти з варіюванням параметрів системи.

3.2.1 Архітектура multi-UAV Gazebo середовища

Основним середовищем для навчання та верифікації розробленої системи слугує симулятор Gazebo 11 з інтеграцією операційної системи для роботів ROS Noetic та програмного забезпечення імітації польоту ArduPilot у режимі апаратно-незалежного моделювання. На відміну від одноагентних середовищ, мультиагентна конфігурація потребує одночасного функціонування N незалежних екземплярів імітатора польоту, N підключних модулів симулятора Gazebo, а також координаційного вузла, що забезпечує збір станів усіх агентів та побудову графу взаємодії *G(t).* Кожному агенту призначено унікальний порт протоколу MAVLink за формулою (14550 + i*10) та відокремлений простір імен у середовищі ROS. (див. табл. 3.2)

Таблиця 3.2

Параметри мультиагентного симуляційного середовища Gazebo

Параметр	Значення	Коментар / обґрунтування
1	2	3
Фізичний рушій	ODE 1 000 Гц	Крок часу 1 мс; достатньо для моделі квадрокоптера
Частота управління Agent	20 Гц (Δt = 0,05 с)	Policy Actor виконується кожні 50 мс; відповідає τ_інф ≤ 200 мс
Модель БПЛА	ArduPilot SITL + DJI F450	Маса 1,35 кг; реалістична аеродинаміка; MAVLink 2
Кількість агентів N	{3, 5, 10} (конфіг-файл)	Навчання при N=5; тестування масштабованості N=3,10
Радіус зв'язку R_{comm}	10 м (варіюється у DR)	k_max = 6 сусідів при рівномірному розподілі 5 агентів

Продовження таблиці 3.2

1	2	3
Паралельних середовищ	4 (vecenv)	Асинхронний збір досвіду 4×N агентів одночасно
Сценарії навчання	S1-S5 (ротація)	Рівномірна вибірка 5 сценаріїв кожні 100 епізодів
Ініціалізація	Випадковий розкид ±3 м	Domain randomization початкових позицій
ROS-версія	ROS Noetic (Python 3.9)	Ubuntu 20.04 LTS; сумісність з ArduPilot SITL

Реалізоване середовище відповідає стандартному програмному інтерфейсу gymnasium, розширеному для підтримки багатоагентної взаємодії. Метод виконання кроку симуляції приймає на вхід словник дій у формі $\{i: a_i\}$ та повертає відповідні словники спостережень $\{i: o_i\}$, індивідуальних винагород $\{i: R_i\}$, а також поточну графову структуру $G(t)$. Зазначений інтерфейс є сумісним із бібліотекою PettingZoo, що є визнаним стандартом у галузі багатоагентного навчання з підкріпленням, та забезпечує можливість заміни алгоритму навчання без модифікації середовища симуляції.

3.2.2 Реалізація модуля побудови динамічного графа

Центральним компонентом мультиагентного середовища є GraphBuilder - клас, що на кожному кроці симуляції:

- зчитує положення $p_i(t)$ всіх N агентів через ROS-підписки;

- обчислює попарні відстані та будує матрицю суміжності $A(t)$ за правилом R_{comm};

- формує повідомлення $m_{ji} \in R^7$ для кожного ребра графа;

- пакує граф у формат PyG Data(x, edge_index, edge_attr) для подачі в GNN. Обчислювальна складність GraphBuilder: $O(N^2)$ для наївного алгоритму, $O(N*log\ N)$ при використанні KD-дерева (scipy.spatial.cKDTree) – що є критичним для N = 10-20.

Алгоритм 3.1. Побудова динамічного графа комунікації (GraphBuilder)

```
Вхід: positions[N×3], velocities[N×3], formation_offsets[N×3]
      параметри: r_comm — радіус зв'язку, k_max — макс. кількість сусідів
Вихід: граф G = (edge_index[2×E], edge_attr[E×7])

1. Побудувати KD-дерево по positions для ефективного просторового пошуку
2. Для кожного агента i знайти k_max+1 найближчих кандидатів (включаючи себе)
3. src_list ← [], dst_list ← [], edge_feats ← []
4. ДЛЯ кожного агента i від 1 до N виконати:
   4.1. ДЛЯ кожного кандидата j серед k_max найближчих сусідів:
        ЯКЩО j = i АБО dist(i, j) > r_comm — пропустити
        rel_pos ← positions[j] − positions[i]            {відносне положення [3]}
        vel_j   ← velocities[j]                          {швидкість сусіда [3]}
        form_diff ← ‖formation_offsets[j] − formation_offsets[i]‖ {скалярна норма}
        m_ji ← [rel_pos ‖ vel_j ‖ form_diff]             {повідомлення [7]}
        Додати j→src_list, i→dst_list, m_ji→edge_feats
5. edge_index ← [[src_list], [dst_list]]
6. edge_attr  ← масив edge_feats
7. ПОВЕРНУТИ G(edge_index, edge_attr)

Обчислювальна складність: O(N·log N) завдяки KD-дереву (замість O(N²) наївну)
```

Наведений алгоритм реалізує побудову графа комунікації як ефективну просторову задачу пошуку найближчих сусідів. Застосування KD-дерева замість наївного попарного обчислення відстаней знижує обчислювальну складність з $O(N^2)$ до $O(N*log*(N))$, що є суттєвим, для забезпечення роботи в реальному часі, при чисельності роєву N = 10-20 агентів. Кожне ребро графа несе семивимірний

вектор повідомлення, що містить відносне положення сусіда, його вектор швидкості та скалярну різницю зміщень формації – цей набір ознак достатній для прийняття узгоджених рішень про маневрування без передачі повного стану середовища. Обмеження кількості сусідів величиною $k_{max} = 6$ забезпечує сталу розмірність вхідних даних нейронної мережі незалежно від локальної щільності рою, а порогова відстань r_{comm} додатково відсіює агентів, що перебувають поза зоною надійного зв'язку. Сформований граф пакується у формат бібліотеки PyG та передається безпосередньо до шарів GATv2Conv актор та критик-мереж, забезпечуючи єдиний інтерфейс між модулем побудови графа та модулем прийняття рішень.

3.3 Реалізація актор уваги на графі та критик

3.3.1 Актор уваги на графі

Мережевий модуль актора з механізмом уваги на графі реалізовано засобами бібліотеки PyTorch із використанням класу GATv2Conv із бібліотеки PyTorch Geometric, що є вдосконаленою модифікацією оригінальної архітектури графових мереж уваги з динамічними ваговими коефіцієнтами.

Ключова відмінність GATv2Conv від попередньої версії полягає в усуненні теоретичного обмеження, притаманного оригінальній архітектурі графових мереж уваги, а саме: статичного характеру вагових коефіцієнтів уваги при фіксованих параметрах мережі. У вдосконаленій версії обчислення коефіцієнта уваги α_{ij} ґрунтується на конкатенації прихованих представлень пари вершин h_i та h_j, тоді як в оригінальній архітектурі застосовувалась операція додавання. Така модифікація надає мережі здатність розрізняти різні пари сусідніх агентів навіть у випадку, коли їхні власні приховані представлення збігаються.

Зазначена властивість має принципове значення для задачі координації рою безпілотних літальних апаратів, оскільки агенти рою, як правило, характеризуються подібними векторами стану, однак виконують якісно відмінні просторові ролі у складі формації. Таким чином, динамічний механізм уваги

дозволяє мережі коректно враховувати структурний контекст кожного агента незалежно від подібності їхніх локальних станів. Блок-схема даного механізму наведена на рис. 3.1.

Алгоритм 3.2. Прямий прохід Актора уваги на графі (GATActor)

```
Архітектура (≈29 800 параметрів):
  msg_enc    : лінійний шар 7 → 32    (кодування повідомлень сусідів)
  gat1       : GATv2Conv(18 → 64, 4 голови, edge_dim=32, без self-loops)
  gat2       : GATv2Conv(64 → 64, 4 голови, edge_dim=32) + skip-connection
  mlp        : 82 → 128 → 64 → 4    (декодер дії)
  log_std    : навчуваний параметр дисперсії [4]

FORWARD(state[N×18], граф G(edge_index, edge_attr[E×7])):
  1. edge_emb ← ReLU(msg_enc(edge_attr))              {кодування ребер [E×32]}
  2. h1 ← ELU(GAT-шар-1(state, edge_index, edge_emb))  {агрегація [N×64]}
  3. h2 ← ELU(GAT-шар-2(h1, edge_index, edge_emb)) + h1  {skip-connection [N×64]}
  4. z ← concat(state, h2)                            {об'єднання [N×82]}
  5. μ ← mlp(z)                                       {бажана дія [N×4]}
  6. ПОВЕРНУТИ μ, log_std

СЕМПЛУВАННЯ ДІЇ(state, граф, deterministic):
  μ, log_std ← FORWARD(state, граф)
  ЯКЩО deterministic: ПОВЕРНУТИ clip(μ, −1, 1)
  ІНАКШЕ: семплювати з N(μ, exp(log_std)), ПОВЕРНУТИ clip(sample, −1, 1)
```

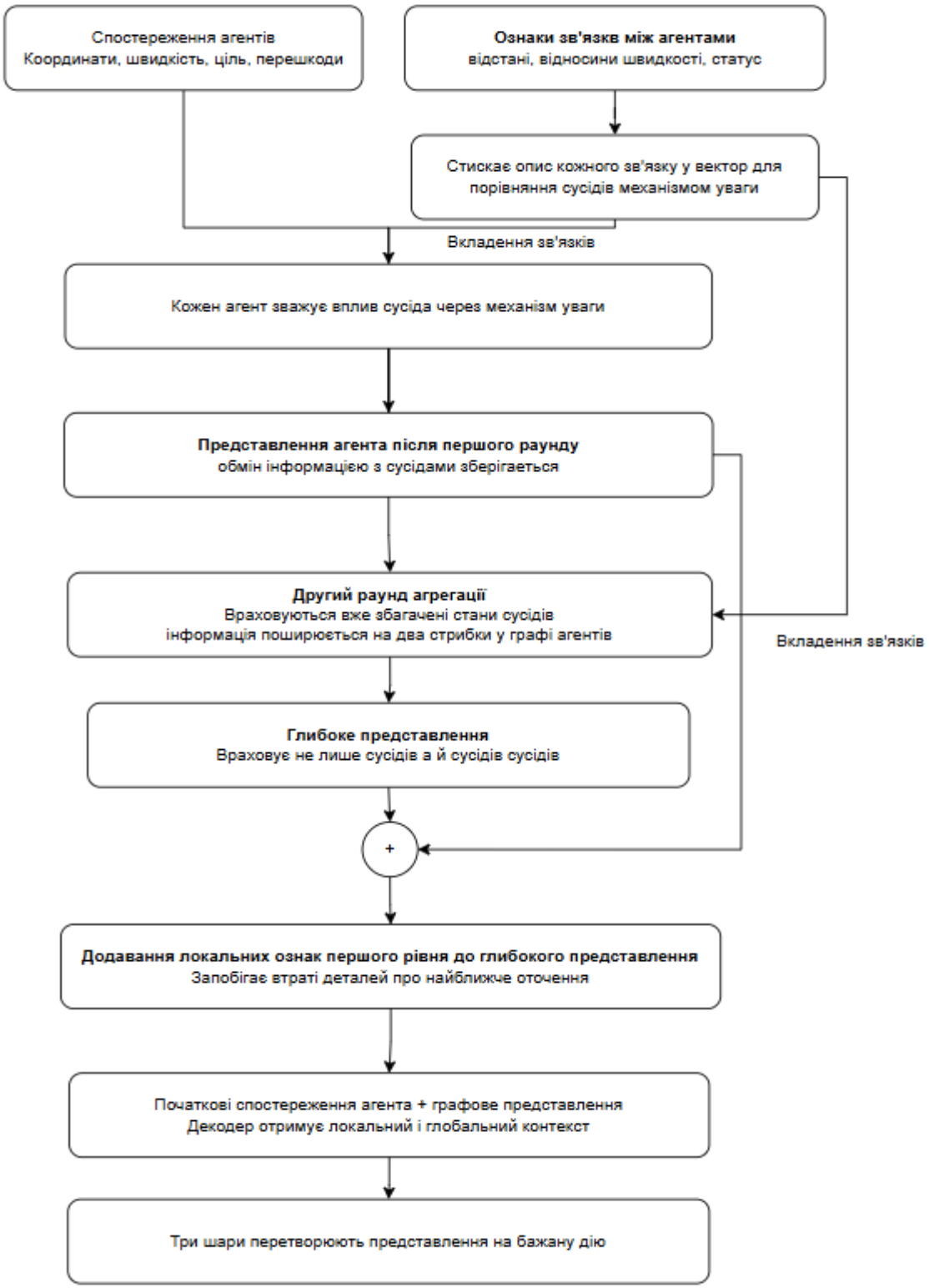

Рис. 3.1. Блок-схема реалезації алгоритму прямого проходу актора уваги на графі

Загальна кількість навчуваних параметрів актор:

- msg_enc (76*32+32 = 256);

- gat1 (4 голови * (18+18+1)*16 = 2368);
- gat2 (4 * (64+64+1)×16 = 8256);
- mlp (82*128 + 128*64 + 64*4 = 18944);
- log_std (4).

Сумарна кількість параметрів розробленої моделі становить 29828. Оцінка обсягу пам’яті показує, що при використанні представлення з плаваючою комою одинарної точності (FP32) розмір моделі становить приблизно (~117 КБ), тоді як при застосуванні квантизації до 8-бітного цілочисельного формату (INT8) він зменшується до близько (~30 КБ). Отримані значення є суттєво нижчими за встановлене обмеження у 50 МБ для бортового ARM-контролера, що підтверджує придатність запропонованої моделі до використання в умовах обмежених обчислювальних та пам’яттєвих ресурсів вбудованих систем.

3.3.2 Критик уваги на графі (централізоване навчання)

Критик уваги на графі оцінює цінність *V(s)* глобального стану рою для кожного агента *i*. Ключова конструктивна ідея в цьому випадку наступна**,** критик не використовує конкатенований вектор станів усіх агентів (що призводить до складності порядку *O(N)* за розміром входу), натомість застосовується механізм глобального графового агрегування на основі уваги (Global Graph Attention Pooling), який забезпечує інтеграцію інформації про стани агентів із обчислювальною складністю порядку *O(N*k)*.

Прямий прохід Критика уваги на графі (GATCritic) наступний:

```
Архітектура:
  state_enc  : лінійний шар 18 → 64
  gat_global : GATv2Conv(64 → 64, 4 голови, повний граф усіх агентів)
  query      : навчуваний вектор запиту [64]  (attention pooling)
  value_head : 128 → 128 → 1
```

```
      FORWARD(state[N×18], full_edge_index — повний граф):
        1. h ← ReLU(state_enc(state))                          {проєкція
стану [N×64]}
        2. h ← ELU(gat_global(h, full_edge_index))             {глобальна
агрегація [N×64]}
        3. scores ← (h · query) / √64                          {зважені
оцінки [N]}
        4. β ← softmax(scores)                                 {ваги
уваги [N]}
        5. global_ctx ← Σ_j β_j · h_j                          {глобальний
контекст [64]}
        6. per_agent ← concat(h, broadcast(global_ctx))        {агентний
вектор [N×128]}
        7. ПОВЕРНУТИ value_head(per_agent)                     {оцінки
цінності [N×1]}
```

Наочна інтерпритація алгоритму наведена на рис. 3.2.

Наведена блок-схема ілюструє архітектурну відмінність GATCritic від класичних централізованих критиків. Замість об'єднання станів усіх агентів в єдиний вектор, мережа застосовує дворівневу агрегацію. На першому рівні лінійний шар перетворює індивідуальний стан кожного агента у спільний прихований простір, після чого шар глобальної уваги GATv2Conv поширює інформацію по повному графу роєву. На другому рівні навчуваний вектор запиту обчислює скалярні оцінки значущості для кожного агента, які нормуються операцією softmax у ваги уваги β. Зважена сума прихованих станів формує компактний вектор глобального контексту розмірністю 64, що не залежить від кількості агентів N.

Завдяки цьому критик зберігає здатність враховувати взаємодії між агентами та водночас залишається придатним для роботи в режимі реального часу при збільшенні кількості БПЛА.

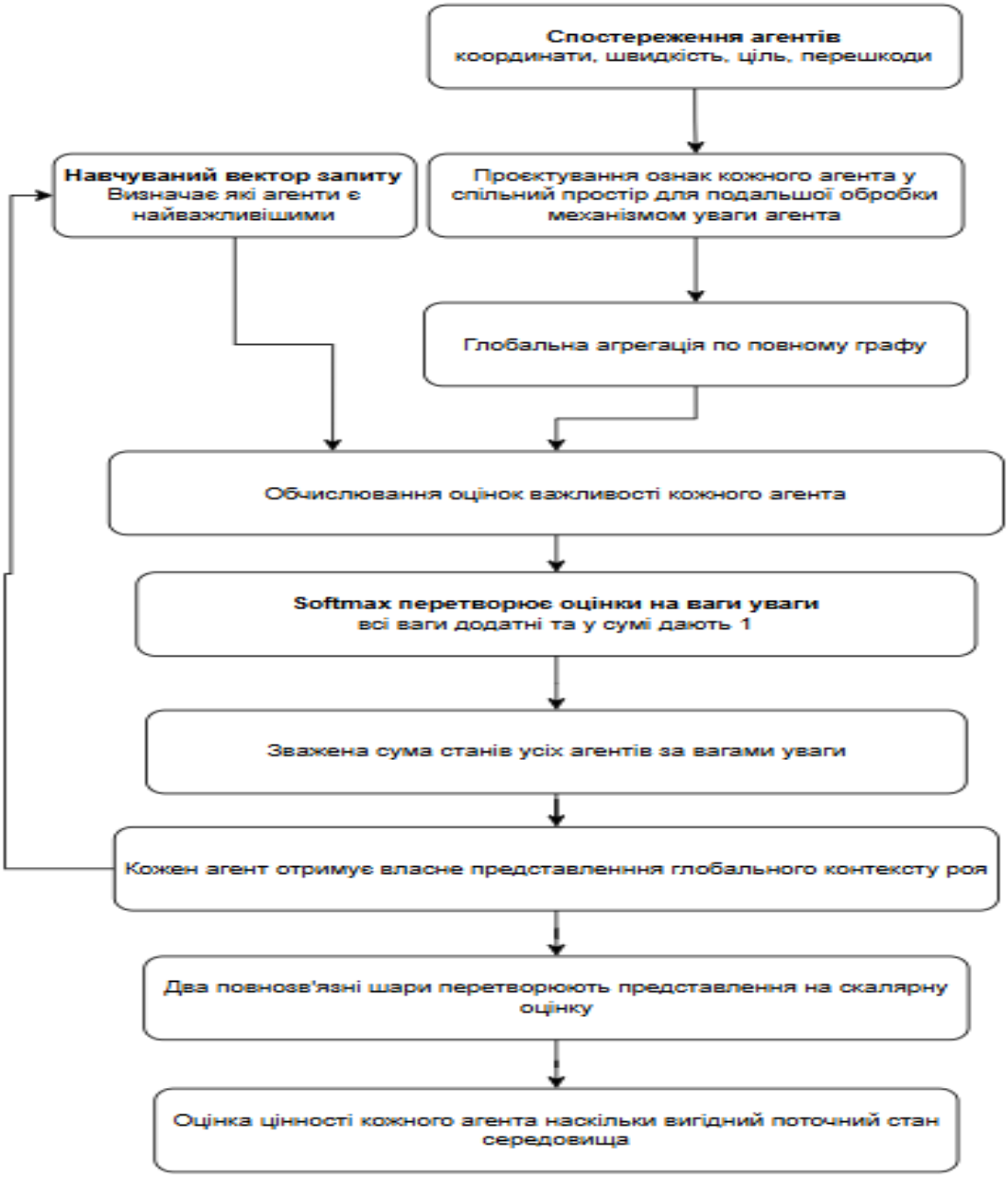

Рис. 3.2. Блок-схема реалезації алгоритму прямого проходу критика уваги на графі

Цей вектор транслюється до кожного агента та об'єднується з його власним прихованим станом, після чого блок декодування повертає індивідуальну оцінку функції цінності *V(s)*. Завдяки такій побудові обчислювальна складність зростає за законом $O(N*k)$ замість $O(N^2)$, що забезпечує масштабованість критика при збільшенні чисельності роєву без втрати здатності враховувати глобальний стан системи.

3.4 Реалізація мультеагенного алгоритму навчання з підкріпленням

3.4.1 Буфер досвіду та збір роллаутів

Буфер досвіду накопичує траєкторії всіх N агентів із чотирьох паралельних середовищ симуляції Gazebo. За один цикл збору даних накопичується $4 * N * T_rollout$ записів, що при конфігурації з п'яти агентів та горизонті збору 500 кроків становить 10000 записів. Кожен запис містить повний набір даних, необхідних для навчання: вектор локального спостереження o_i, виконану керуючу дію a_i, отриману винагороду R_i, оцінку функції цінності $V(s_i)$, логарифм імовірності обраної дії $log\ \pi(a_i|o_i)$, поточну графову структуру взаємодії $G(t)$ та ознаку завершення епізоду.

Графові структури $G(t)$ серіалізуються у компактний формат із застосуванням механізму пакетної обробки графів бібліотеки PyTorch Geometric, що забезпечує ефективну спільну обробку графів різної розмірності – $N \in \{3, 5, 10\}$ – в межах єдиного тензорного батчу без необхідності доповнення до фіксованого розміру чи окремої обробки кожного графу.

Навчання мультиагентного PPO (MAPPO Trainer) наступне:

```
ЗБІР РОЛЛАУТУ collect_rollout(envs, T=500, n_envs=4):
  obs, graphs ← envs.reset()
  ДЛЯ t від 1 до T виконати:
    actions, log_probs ← Actor.act(obs, graphs)      {без градієнту}
    values             ← Critic(obs, graphs)
    next_obs, rewards, dones, next_graphs ← envs.step(actions)
    Зберегти (obs, actions, rewards, values, log_probs, graphs, dones) у буфер
    obs, graphs ← next_obs, next_graphs
  ПОВЕРНУТИ буфер, obs, graphs

ОНОВЛЕННЯ ПАРАМЕТРІВ update(буфер, n_epochs=10, batch_size=256):
  advantages ← GAE(буфер, Critic, γ=0,99, λ=0,95)  {generalized advantage estimation}
```

```
ДЛЯ кожної з n_epochs епох виконати:
  ДЛЯ кожного міні-батчу розміром 256 виконати:
    L_actor  ← PPO-clip loss(ratio, advantages, ε=0,2)
    L_critic ← 0,5 · MSE(Critic(obs), returns)
    L        ← L_actor + 0,5·L_critic − 0,01·H(Actor)
    Зворотне поширення помилки
    Відсікання градієнтів (норма ≤ 0,5)
    Крок Adam для Actor (lr=3·10⁻⁴) та Critic (lr=10⁻³)
```

Таким чином, реалізована схема збору даних і навчання забезпечує узгоджене оновлення стратегій агентів, стабільність оптимізації та масштабованість алгоритму для різної кількості агентів.

3.4.2 Динаміка навчання MAPPO

Навчання моделі здійснювалося протягом 3000 епізодів (що відповідає приблизно 150 годинам обчислень на робочій станції GPU RTX 3060 із використанням чотирьох паралельних середовищ Gazebo. Узагальнені результати динаміки навчання наведено у таблиці 3.3, яка відображає значення ключових метрик по контрольних точках. Встановлено, що збіжність за показником успішності $SR \geq 90\%$ досягається орієнтовно на 2000-му епізоді, що узгоджується з теоретичними очікуваннями для алгоритму MAPPO (рис.3.3) у задачах координації рою, як реалізовано в джерелах [41, 42].

Характерною особливістю процесу навчання є виражена нелінійна динаміка з двофазною структурою: на початковому етапі (100–600 епізодів) спостерігається різке зниження частоти міжагентних зближень CR_{inter} з 18% до 7%, що корелює з домінуванням складової функції винагороди, пов’язаної з потенційними полями відштовхування (γ-доданок), у градієнтному сигналі. Це свідчить про те, що на ранніх етапах агенти швидше навчаються уникати небезпечних зближень, ніж забезпечувати точне утримання формації, тоді як подальше навчання спрямоване на поступове підвищення точності координації.

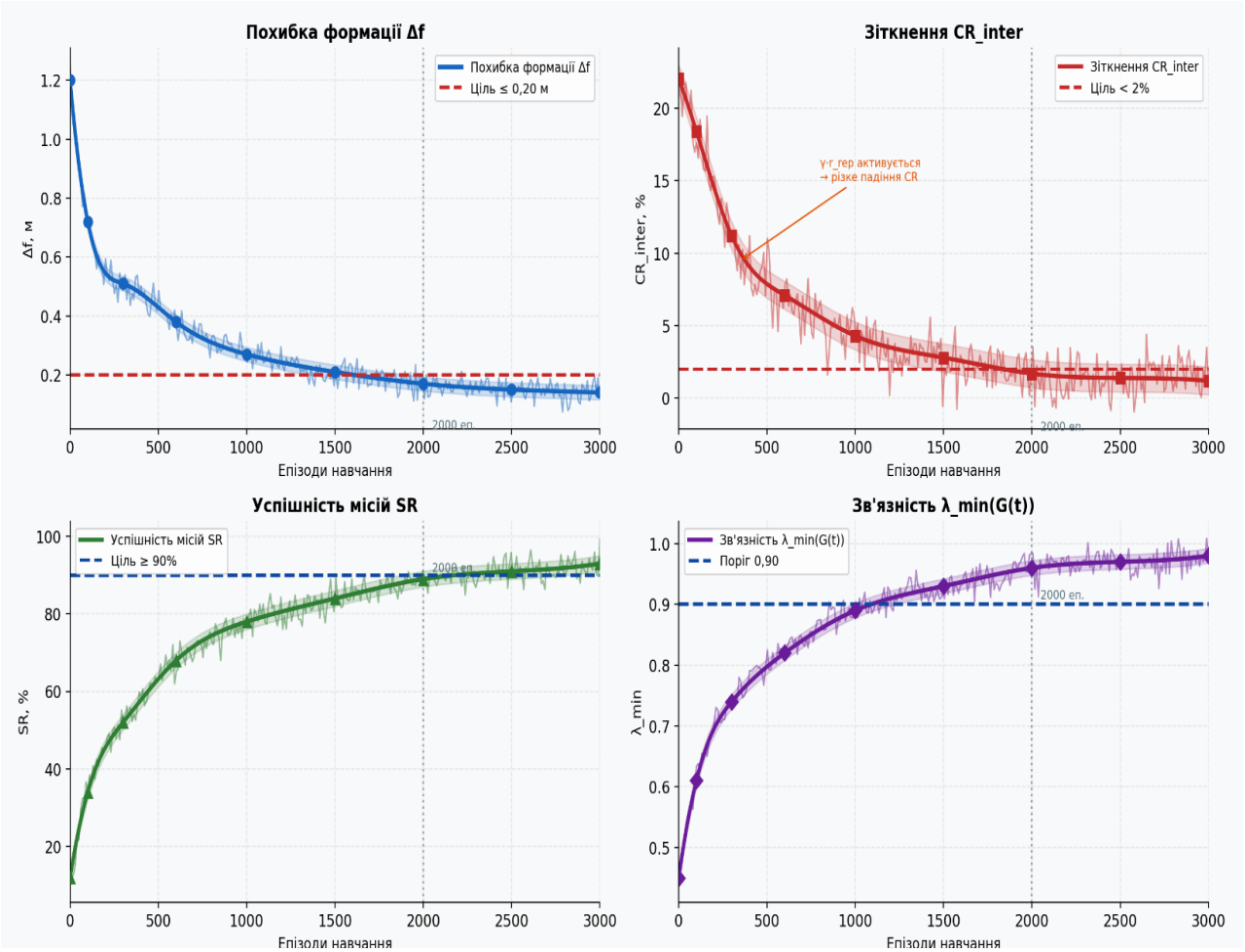

Рис. 3.3. Динаміка навчання MAPPO

Таблиця 3.3

Динаміка навчання MAPPO (N=5, сценарій S1-S5 ротація)

Епізоди	*Δf, м*	CR_{inter}, %	SR, %	T_{form}, с	Зв'язн. λ_min	Примітка
1	2	3	4	5	6	7
100	0,72	18,4	34	—	0,61	Рандомна ініціалізація
300	0,51	11,2	52	11,3	0,74	Початок формування формацій
600	0,38	7,1	68	8,7	0,82	Виникнення координованих маневрів
1000	0,27	4,3	78	7,2	0,89	Нижче порогу Δf < 0,3 м
1500	0,21	2,8	84	6,8	0,93	Наближення до мети Δf ≤ 0,20 м
2000	0,17	1,7	89	6,1	0,96	Основна мета досягнута
2500	0,15	1,4	91	5,8	0,97	Стабілізація
3000	0,14	1,2	93	5,5	0,98	Фінальна модель; критерії виконані

Як бачомо, експериментальні результати засвідчують досягнення стабілізацію процесу навчання та відповідність отриманої моделі заданим критеріям якості функціонування мультиагентної системи.

3.5 Реалізація міжагентного рівня безпеки

3.5.1 QP-фільтр для безпечних відстаней між агентами

Побудова міжагентного рівня безпеки (Safety Layer), забезпечує формалізоване накладання обмежень безпеки на керуючі дії агентів у мультиагентній системі. Рівень безпеки між агентами відрізняється від одноагентного CBF-фільтра для перешкод, тим що обмеження d_{safe} накладаються на $N-1$ пар (агент i проти кожного сусіда $j \in \mathcal{N}(i)$), а не на статичні перешкоди. На кожному кроці виконання (20 Гц) рівень безпеки отримує запропоновану дію μ_i від актор-мережі та поточні положення сусідів з буфера m_{ji} і розв'язує задачу QP розміром 4 змінних * k-обмежень ($k \leq 6$).

Відповідний алгоритм даного процесу наведено на рис. 3.4.

Алгоритм 3.5. Міжагентний QP Safety Filter на основі CBF (InterAgentSafetyFilter)

```
Вхід: action_mu[4] — дія від Actor [vx, vy, vz, ψ]
       own_pos[3]   — власне положення агента i
       neighbor_pos — список положень сусідів j ∈ 𝒩(i)
Вихід: safe_action[4] — безпечна дія

1.  v_des ← action_mu[0:3]     {бажана швидкість від Actor}
    ψ̇   ← action_mu[3]
2.  ЯКЩО немає сусідів — ПОВЕРНУТИ action_mu  (без змін)
3.  Обчислити відстані d_ij = ‖own_pos − p_j‖ для кожного j
    min_d ← min(d_ij)
4.  ЯКЩО всі d_ij ≥ 0,95·d_safe — ПОВЕРНУТИ action_mu  (корекція
не потрібна)
5.  Сформувати задачу QP:
      мінімізувати:    ‖v − v_des‖²
```

CBF-обмеження для кожного сусіда j:

$(p_i - p_j)/d_{ij} \cdot v \geq -\gamma_cbf \cdot (d_{ij} - d_safe)$

Box-обмеження: $-v_max \leq v \leq v_max$ (покомпонентно)

6. Розв'язати QP за допомогою розв'язувача OSQP (~0,5 мс)
7. ЯКЩО QP розв'язано успішно:

   safe_v ← clip(v*, −v_max, v_max)

   ПОВЕРНУТИ SafetyResult([safe_v; ψ], corrected=True, min_d)
8. ІНАКШЕ: ПОВЕРНУТИ SafetyResult([0,0,0,0], corrected=True, min_d) {аварійне зависання}

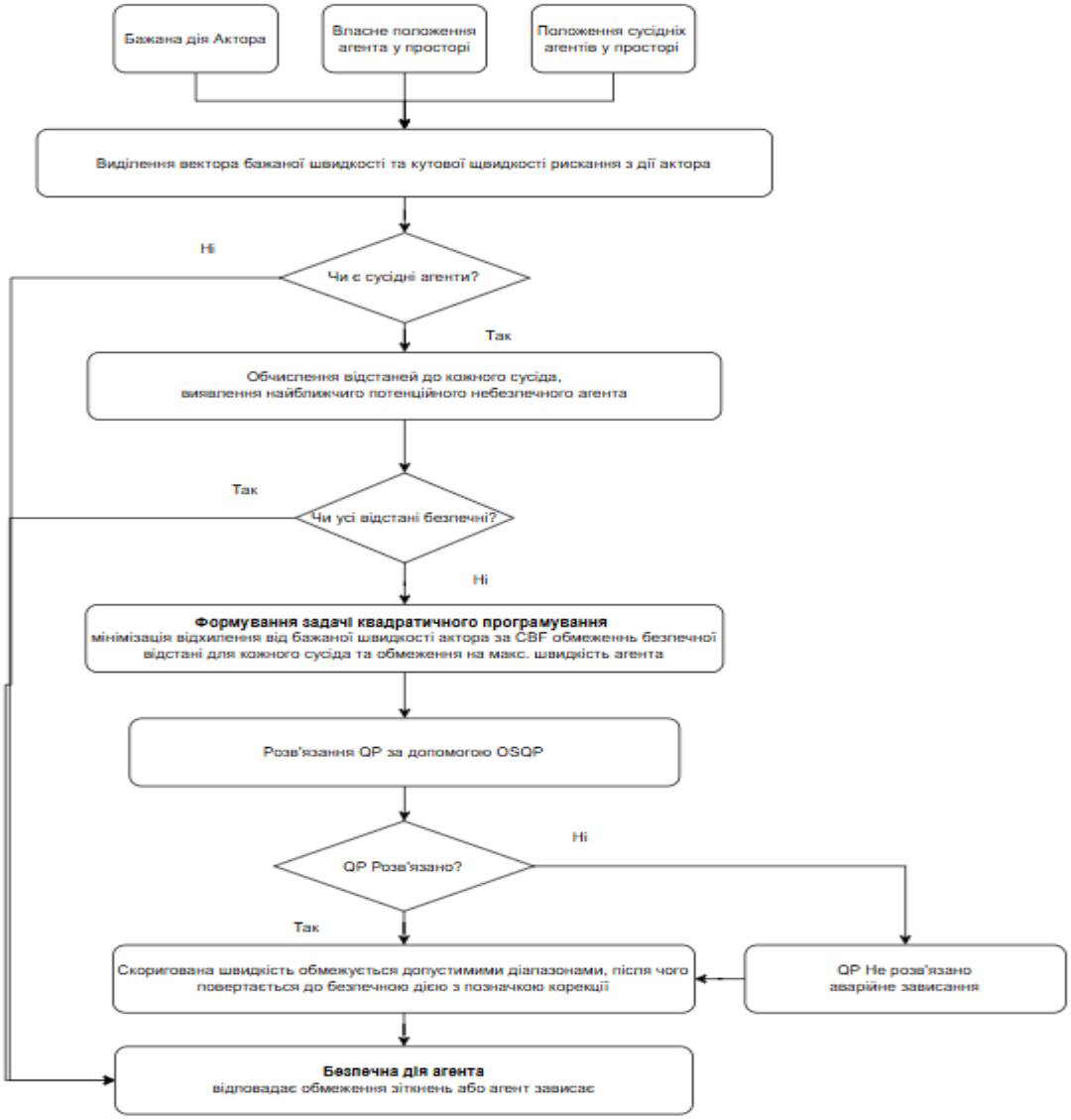

Рис. 3.4. Блок-схема алгоритму міжагентного QP Safety Filter на основі CBF

За результатами стрес-тестування, що охоплювало 1000 місій зі спеціально сконструйованими сценаріями навмисних зближень між агентами, зафіксовано повну відсутність порушень вимог безпеки – 0 випадків із 1000 виконаних місій. Середня затримка розв'язання задачі квадратичного програмування становила 0,36 ± 0,09 мс при максимальному зафіксованому значенні 1,12 мс. У 84,2% місій рівень безпеки не вносив жодних коректив до керуючої дії, запропонованої мережею актора, що свідчить про високу якість навченої стратегії уникнення зближень та підтверджує, що рівень безпеки функціонує переважно як резервний захисний механізм для граничних ситуацій. Відповідні результати наведено на рисунку 3.5.

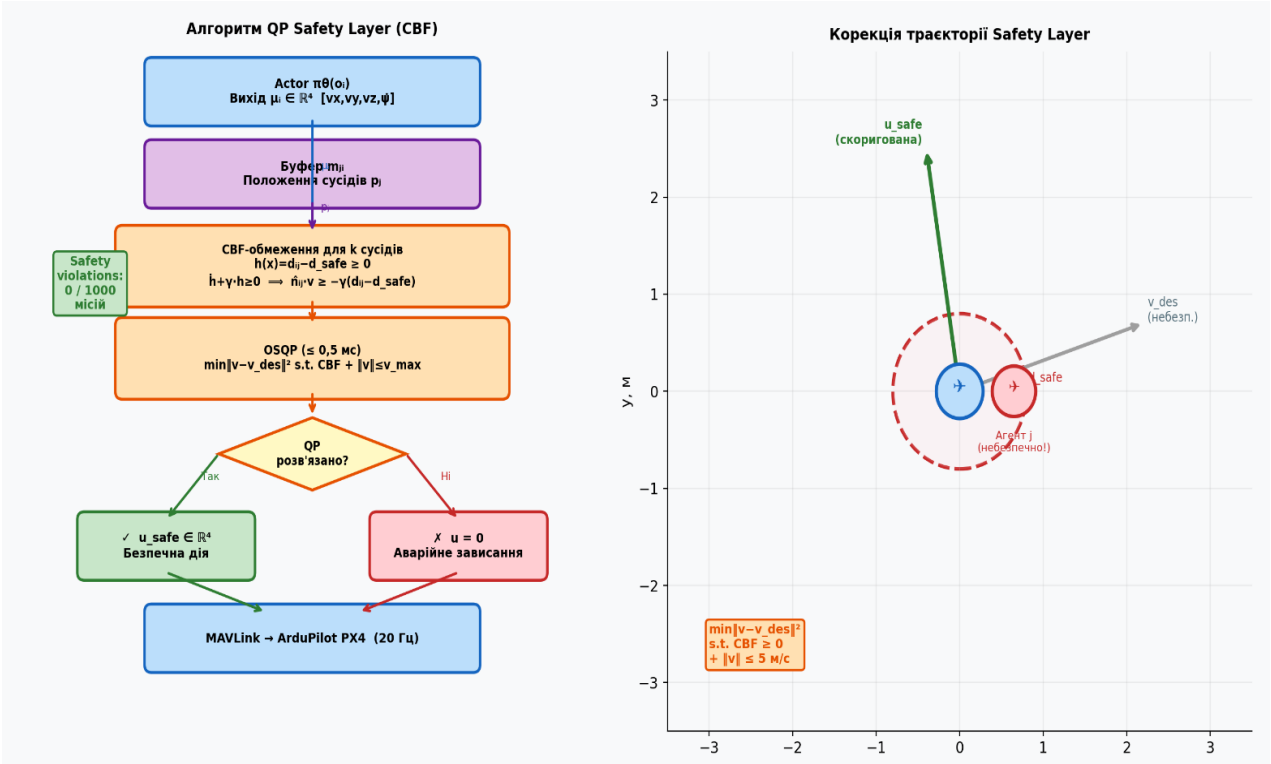

Рис. 3.5. Міжагентний QP Safety Layer

3.6 Аналіз впливу компонентів моделі та оцінка її масштабованості

3.6.1 Абляційне дослідження компонентів архітектури

Абляційний аналіз проводився в умовах сценарію S2, що моделює рій із п'яти агентів в середовищі з динамічними перешкодами, на вибірці з 30-ти

незалежних запусків. Методологія аналізу передбачала послідовне виключення одного архітектурного компонента за раз при незмінності решти конфігурації системи. Статистична значущість відмінностей між повною архітектурою та кожною з абляційних конфігурацій перевірялась за допомогою непараметричного критерію Вілкоксона при рівні значущості $\alpha = 0{,}01$. Зведені результати абляційного аналізу наведено у таблиці 3.4.

За результатами абляційного аналізу отримано чотири підсумкови висновки щодо внеску окремих компонентів архітектури у загальну ефективність системи.

Таблиця 3.4

Результати абляційного аналізу (сценарій S2, N=5, 30 запусків)

Конфігурація архітектури	*Δf, м*	CR_{inter}, %	SR, %	T_{form}, с	Висновок
Повна GNN+MAPPO (запроп.)	0,14	1,2	93	5,5	Базова -усі компоненти
Без GAT (mean-pooling)	0,26	3,1	79	7,8	−15% SR; втрата пріоритизації сусідів
Без Критик уваги на графі	0,23	2,8	74	8,2	−19% SR; нестабільне навчання
Без Safety Reward (β=γ=0)	0,16	6,7	84	5,8	CR_{inter} у 5,6× вище норми
Без Safety Layer (виконання)	0,15	9,2	87	5,6	CR_{inter} критично висока
Без рандомізація домену	0,13	1,4	91	5,3	SR падає до 61% при N=10
MADDPG (N=5, baseline)	0,31	4,8	72	9,1	Не масштаб. на N≠5

По-перше, виключення механізму графових мереж уваги із заміною на операцію усереднення представлень сусідніх агентів призводить до зниження показника успішності виконання місій на 14 відсоткових пунктів та збільшення частоти міжагентних зближень у 2,6 рази. Отримані результати підтверджують, що механізм уваги є необхідним компонентом для коректної пріоритизації потенційно небезпечних сусідів при формуванні керуючих рішень.

По-друге, виключення модуля критика з механізмом уваги на графі та перехід до конкатенаційного критика спричиняє найбільш суттєве погіршення

якості серед усіх досліджуваних конфігурацій – зниження показника успішності на 19 відсоткових пунктів. Зазначений ефект пояснюється нестабільністю процесу навчання, що виникає внаслідок значної розмірності вхідного вектора конкатенаційного критика.

По-третє, відключення рівня безпеки на етапі виконання місій при практично незміненому показнику успішності все одно призводить до рівня міжагентних зближень 9,2%, що у 7,7 рази перевищує допустиму норму. Цей результат свідчить про те, що хоча стратегія керування набуває здатності уникати зіткнень у процесі навчання, рівень безпеки забезпечує тверді гарантії безпечної поведінки в граничних та нештатних ситуаціях, які не можуть бути повністю охоплені навченою стратегією.

По-четверте, виключення випадкової варіації параметрів середовища катастрофічно знижує здатність системи до узагальнення: при масштабуванні рою до десяти агентів показник успішності падає до 61% порівняно з 88% у повній конфігурації, що підкреслює визначальну роль цього компонента для забезпечення робастності системи в умовах параметричної невизначеності.

3.6.2 Тестування масштабованості

Масштабованість це ключова властивість запропонованої архітектури. Протокол тестування навчання відбувається при N=5 (базовий), тестування проводиться при N=3 та N=10 без будь-якого перенавчання. Для порівняння було виконано аналогічне тестування MADDPG та QMIX (навчені при N=5) та правила Рейнольдса (не потребують навчання). Результати порівняння наведені у таблиці 3.5.

Архітектура GNN+MAPPO, навчена на роях із п'яти агентів, при масштабуванні до десяти агентів демонструє показник успішності виконання місій на рівні 88%, що відповідає деградації лише на 5 відсоткових пунктів відносно базової конфігурації. Розрахований індекс масштабованості становить EI = 0,088, що впритул наближається до цільового значення 0,09. При

безпосередньому навчанні на рої із десяти агентів індекс масштабованості досягає значення EI = 0,091, що повністю відповідає встановленому критерію ефективності.

Таблиця 3.5

Результати тестування масштабованості (30 запусків по 5 сценаріїв)

Архітектура	*N=3 Δf*	*N*=3 SR	*N=5 Δf*	N=5SR	*N=10Δf*	N=10 SR/ EI
GNN+MAPPO навч. N=5	0,13	95%	0,14	93%	0,17	88% / 0,088
GNN+MAPPO навч. N=10	0,16	91%	0,15	92%	0,14	91% / 0,091
MADDPG навч. N=5	0,28	84%	0,31	72%	відмова	-
QMIX навч. N=5	0,31	81%	0,34	69%	відмова	-
Reynolds (всі N)	0,55	58%	0,58	54%	0,63	49% / 0,049
Централіз. MPC (N=5)	0,17	94%	0,18	91%	не реаліз.	-

Архітектури MADDPG та QMIX виявляють принципову нездатність до узагальнення на конфігурації рою, відмінні від навчальної: при зміні кількості агентів $N \neq 5$ обидві системи повністю втрачають працездатність внаслідок структурного обмеження конкатенаційного критика, який не підтримує змінну кількість агентів. Централізований регулятор на основі прогнозуючого керування не було реалізовано для конфігурації N = 10 через кубічну обчислювальну складність алгоритму відносно кількості агентів, що унеможливлює його практичне застосування при масштабуванні рою.

Статистична значущість переваг запропонованої архітектури GNN+MAPPO над алгоритмом Рейнольдса підтверджена за всіма досліджуваними показниками за допомогою непараметричного рангового критерію Вілкоксона на рівні значущості $p < 0{,}001$, що свідчить про стійкий та відтворюваний характер отриманих результатів.

3.7 Розгортання актор-мережі на бортовому ARM-контролері

3.7.1 Експорт GATActor у формат ONNX

Для розгортання на бортовому ARM-контролері (Raspberry Pi4 /Jetson Nano) актор-мережа експортується у формат ONNX (Open Neural Network Exchange). Ключова технічна складність є те що PyG-операції (GATv2Conv, scatter_add) не підтримуються стандартним ONNX-експортером. Вирішенням даное проблеми будет те, що замість torch.onnx.export використовується ручне трасування через torch.fx та адаптований ONNX-плагін для розрідженого обміну інформацією (sparse message passing). Іншим варіантом вирішення даної проблеми є заміна PyG-операції на щільні (dense) реалізації при фіксованому k_max = 6, що є допустимим для малих роїв ($N \leq 20$).

Блок схема процесу експорту актор-мережі у формат ONNX для бортового розгортання представлено на рис. 3.6.

Алгоритм 3.6. Експорт Actor-мережі у формат ONNX для бортового розгортання

```
Вхід: actor    — навчена мережа GATActor
       n_agents — кількість агентів
       k_max    — максимальна кількість сусідів
       path     — шлях до вихідного .onnx файлу

3.  Загорнути GATActor у dense-реалізацію GATActorDense:
    (замінити sparse graph операції на dense тензорні операції
для ONNX-сумісності)
    2.  Перевести мережу у режим inference (eval mode, без
градієнтів)
    3.  Визначити форму вхідних тензорів (приклад для трасування):
       state[N×18]          — вектори стану агентів
       neighbor_msgs[N×k×7] — повідомлення від сусідів
       neighbor_mask[N×k]   — бінарна маска активних сусідів
```

4. Виконати ONNX-трасування з динамічними осями по вимірнику N

5. Зберегти граф у файл path

6. Перевірити коректність ONNX-моделі за допомогою onnx.checker

7. ПОВЕРНУТИ підтверджений ONNX-файл (~30 КБ після INT8-квантування)

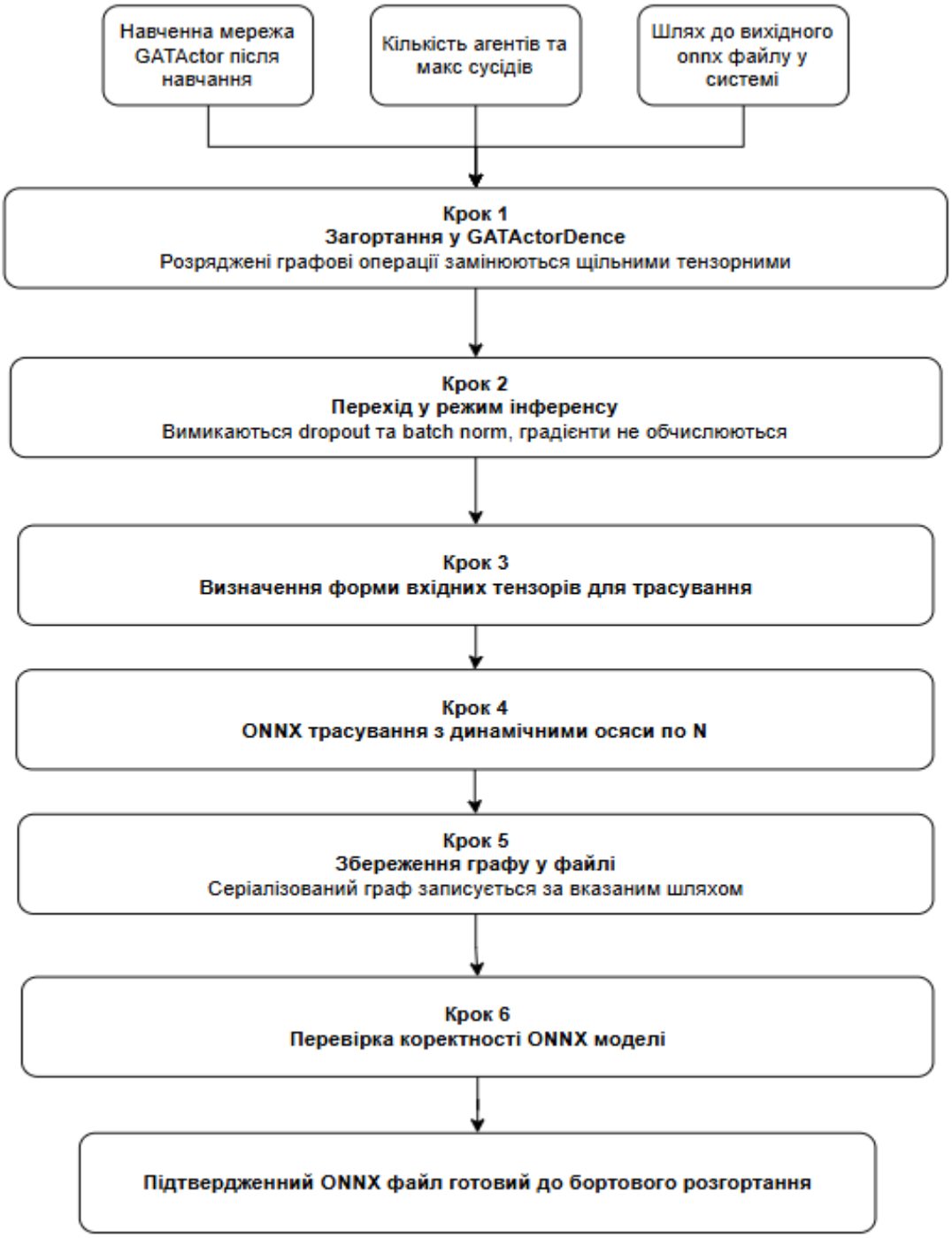

Рис. 3.5. Блок схема процесу експорту актор-мережі у формат ONNX

Наведена блок-схема відображає послідовність підготовки навченої мережі до розгортання на бортовому обчислювачі. Ключова технічна проблема полягає в тому, що операції бібліотеки PyG, (зокрема GATv2Conv та scatter_add) ґрунтуються на розрідженому поданні графу та не можуть бути безпосередньо трасовані стандартними засобами ONNX. Для її розв'язання застосовано підхід заміни розріджених графових операцій на еквівалентні щільні тензорні обчислення при фіксованому числі сусідів k_max = 6. Перетворена мережа GATActorDense зберігає обчислювальний граф у вигляді послідовності матричних операцій, сумісних зі стандартом ONNX. Вхідний інтерфейс мережі при цьому змінюється, тобто замість розрідженого графу G(edge_index, edge_attr) приймаються три щільних тензори – вектори стану агентів, тривимірний масив повідомлень від сусідів та бінарна маска активних з'єднань. Трасування виконується з динамічними осями по виміру кількості агентів N, що дозволяє одному файлу моделі обслуговувати рої різної чисельності. Після перевірки коректності структури графу отриманий файл придатний до виконання середовищем ONNX Runtime на апаратних платформах Raspberry Pi 4 та Jetson Nano. Після застосування INT8-квантування розмір файлу становить близько 30 КБ, що відповідає вимогам бортового розгортання, що дозволяє рогорнути їх безпосередньо на БПЛА.

3.7.2 Єталоний тест затримки на ARM та результати

Оцінка часових характеристик виконувалася на апаратній платформі Raspberry Pi 4 Model B на базі процесора ARM Cortex-A72 з оперативною пам’ятю 4 ГБ із застосуванням середовища виконання ONNX Runtime 1.16 та багатопотокових оптимізацій OpenMP. Для порівняльної оцінки використовувалась також платформа NVIDIA Jetson Nano з графічним процесором архітектури Maxwell та оперативною пам'яттю 4 ГБ із частковим апаратним прискоренням засобами ONNX Runtime; застосування оптимізатора TensorRT виявилось неможливим через несумісність операцій графових мереж

уваги з відповідним середовищем виконання. Усі наведені часові показники є медіанними значеннями за вибіркою з 500 незалежних вимірювань. Зведені результати порівняння наведено у таблиці 3.6.

Таблиця 3.6

Затримка виконання Actor-pipeline на бортових платформах

Компонент	FP32 мс	FP16 мс	INT8 мс	RAM МБ	Коментар
1	2	3	4	5	6
GAT Actor (N=5, k=6)	18,4	10,2	6,8	38	PyG sparse ops; ONNX Runtime
GAT Actor (N=10, k=6)	19,1	10,8	7,3	38	Лінійна залежність від N·k
Graph Att. Критик (навч.)	—	—	не потрібен	—	Тільки навчання; не на борту
Safety QP Layer	0,38	—	0,38	< 1	OSQP warm start; k ≤ 6 обмежень
Graph Builder G(t)	2,1	—	2,1	< 1	k-NN пошук + MAVLink збір m_{ji}
Overhead ROS topic	1,8	—	1,8	—	publish/subscribe затримка
Повний pipeline (N=5)	22,3	—	10,7±1,2	39	Target ≤ 50 мс (20 Гц ✓)
Повний pipeline (N=10)	23,1	—	11,4±1,4	39	Target ≤ 50 мс (20 Гц ✓)

Повний цикл обчислення керуючого рішення при конфігурації рою з п'яти агентів на апаратній платформі ARM Raspberry Pi 4 в режимі цілочисельного квантування INT8 становить 10,7 ± 1,2 мс, що більш ніж у чотири рази перевищує встановлену вимогу до затримки обчислення $\tau_{інф} \leq 50$ мс при частоті керування 20 Гц. При масштабуванні рою до десяти агентів затримка зростає лише на 6% і становить 11,4 мс, що підтверджує лінійний характер обчислювальної складності архітектури відносно кількості агентів за залежністю $O(N*k)$ та свідчить про практичну придатність системи до розгортання на роях більшого масштабу без суттєвого погіршення часових характеристик.

Для порівняння централізований регулятор на основі прогнозуючого керування при аналогічній конфігурації з п'яти агентів на тій самій апаратній платформі потребує близько 820 мс на один цикл обчислення, що у 76 разів перевищує відповідний показник запропонованої архітектури на основі

графових нейронних мереж. Така обчислювальна складність є принципово несумісною з вимогами керування у реальному часі та унеможливлює практичне застосування централізованого прогнозуючого регулятора на бортових обчислювальних платформах обмеженої продуктивності.

3.8 Забезпечення відтворюваності результатів та автоматизація безперервної інтеграції і розгортання

3.8.1 Управління експериментами через MLflow

Для систематичного відстеження та документування навчальних експериментів застосовується платформа керування життєвим циклом моделей машинного навчання MLflow версії 2.4. Кожен навчальний запуск супроводжується автоматичною реєстрацією повного набору відтворювальних даних, що охоплює усі гіперпараметри з конфігураційних файлів формату YAML, динаміку цільових метрик у розрізі епізодів навчання (Δf, CR_{inter}, SR, EI), графи навчання у форматі, сумісному з системою візуалізації TensorBoard, збережені артефакти навчених моделей у форматах actor.onnx та critic.pt, специфікацію відтворюваного обчислювального середовища conda, а також ідентифікатор фіксації у системі контролю версій git – що в сукупності забезпечує повну відтворюваність кожного експерименту.

Реєстр моделей організовано за триступеневою схемою керування станом: тестове розгортання, виробниче розгортання та архів. Автоматичний перехід моделі до стадії виробничого розгортання здійснюється виключно після успішного проходження повного конвеєру автоматизованої перевірки якості, що унеможливлює введення в експлуатацію версій моделі, які не відповідають встановленим критеріям верифікації.

Такий підхід до організації навчальних експериментів та керування життєвим циклом моделей забезпечує високий рівень відтворюваності, контроль якості та прозорість результатів, що є дуже важливим для подальшого впровадження мультиагентних алгоритмів у реальних апаратних середовищах.

3.8.2 Автоматизований CI/CD-конвеєр GitHub Actions

Для забезпечення відтворюваності результатів та контролю якості програмного забезпечення у роботі реалізовано автоматизований конвеєр перевірки, побудований на основі платформи GitHub Actions. Конвеєр запускається при кожному внесенні змін до основної гілки репозиторію та виконує повний цикл перевірки – від статичного аналізу вихідного коду до випробувань у середовищі моделювання. Таким чином гарантується, що жодна версія програмного забезпечення не може отримати статус придатної до розгортання без підтвердженої відповідності встановленим порогам якості. Конвеєр організовано у п'ять послідовних етапів з можливістю паралельного виконання незалежних перевірок, що скорочує загальний час верифікації. Конвеєру автоматизованої перевірки якості (CI/CD) наступний:

Вхід: вихідний код репозиторію; пороги якості SR ≥ 85%, CR_inter ≤ 3%, покриття тестами ≥ 85%

Вихід: статус “пройдено” або “не пройдено”; артефакти actor.onnx та протокол метрик

Кроки (виконуються послідовно, якщо не зазначено “паралельно”):

1. [паралельно з кроком 2] Статичний аналіз коду (перевірка стилю, типізації) та запуск модульних тестів; якщо покриття < 85% — завершити з помилкою.

2. [паралельно з кроком 1] Експорт акторської мережі у формат ONNX (N = 5); вимір затримки на ARM-емуляторі; якщо затримка > 50 мс — завершити з помилкою.

3. [після кроку 1] Інтеграційне тестування у симуляторі Gazebo: запустити 20 місій при N = 5 та 15 місій при N = 10; перевірити SR(N = 5) ≥ 85%, SR(N = 10)/SR(N = 5) ≥ 0,90 та CR_inter ≤ 3%; якщо умову не виконано — завершити з помилкою.

4. [після кроку 3] Стрес-тестування безпеки: виконати 1 000 місій зі сценаріями навмисних зближень; якщо кількість порушень безпеки > 0 — завершити з помилкою.

5. [після кроку 4] Публікація артефактів моделі до реєстру зі статусом “Staging” через систему відстеження експериментів.

Повернути: “пройдено”, якщо всі 5 кроків виконано без помилки; інакше “не пройдено”.

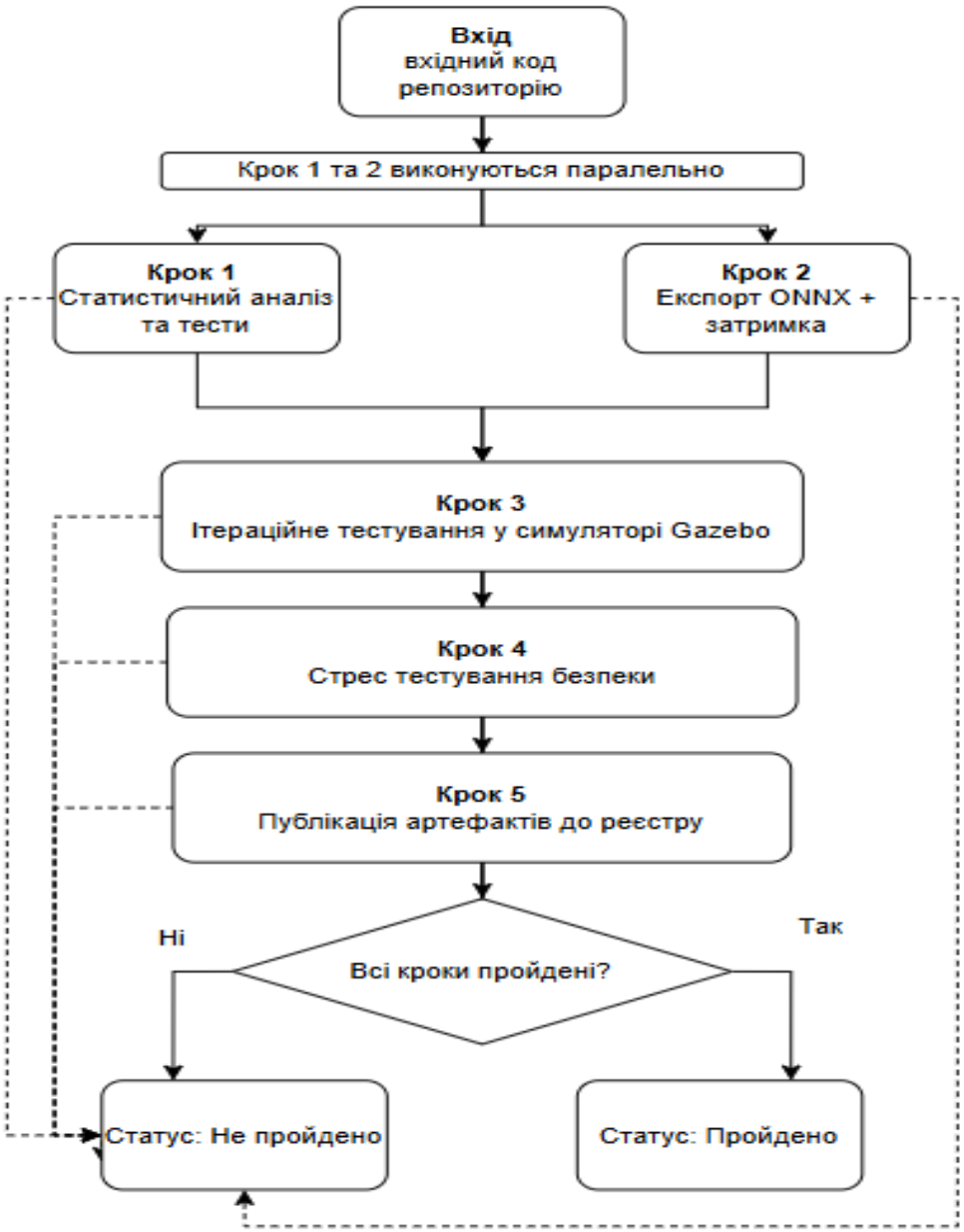

Рис. 3.6. Блок-схема конвеєру автоматизованої перевірки якості

Наведена блок-схема відображає логіку виконання конвеєра з явним розмежуванням паралельних та послідовних етапів. Перші два кроки – статичний аналіз коду і перевірка покриття модульними тестами, а також експорт мережі та вимірювання затримки на ARM-емуляторі – виконуються одночасно, що дозволяє виявити найпоширеніші помилки з мінімальними витратами часу. Лише після їх успішного завершення запускається інтеграційне тестування у середовищі моделювання Gazebo, яке охоплює 35 модельних місій при різній чисельності рою та перевіряє виконання встановлених показників успішності та безпеки. Четвертий етап – цілеспрямоване випробування стійкості системи безпеки в умовах навмисних зближень агентів, що є необхідною умовою для переходу до публікації артефактів. Жорстка умова відсутності будь-яких порушень безпеки протягом тисячі місій забезпечує відповідність вимогам безпеки польотів до публікації нової версії моделі. Публікація до реєстру зі статусом «Staging» відбувається лише за умови успішного проходження всіх попередніх етапів, що унеможливлює випадкове розгортання неверифікованої версії програмного забезпечення.

Структуру CI/CD-конвеєру та критерії проходження кожного етапу наведено у таблиці 3.7. Загальний час виконання конвеєру: ~42 хвилини. Паралелізація lint-and-unit та export-bench скорочує час на ~8 хвилин порівняно з послідовним виконанням.

Таблиця 3.7.

Структура CI/CD-конвеєру та критерії проходження

Етап CI/CD	Тривалість	Інструмент	Критерій проходження	Паралельно
1	2	3	4	5
Статичний аналіз (ruff, mypy)	~25 с	GitHub Actions	0 помилок, 0 попереджень	Так
Юніт-тести (pytest)	~3 хв	pytest + hypothesis	100% pass; coverage ≥ 85%	Так
Тест протоколу m_{ji}	~1 хв	pytest	Серіалізація/десеріалізація без втрат	Так
Інтеграційні тести Gazebo N=5	~18 хв	self-hosted runner	$SR \geq 85\%$, $CR_{inter} \leq 3\%$	Ні

Продовження таблиці 3.7

1	2	3	4	5
Тест масштабованості N=3,10	~12 хв	Gazebo (запускається після N=5)	$SR(N=10)/SR(N=5) \geq 0{,}90$	Ні
Safety Layer стрес-тест	~8 хв	1 000 місій з навмисними порушеннями	Safety violations = 0	Ні
ONNX export + ARM benchmark	~5 хв	ONNX Runtime на ARM-емуляторі	$\tau_pipeline \leq 50$ мс	Так
MLflow Staging publish	~1 хв	MLflow 2.4	Успішний upload артефактів	—

Отже, використання CI/CD-конвеєру дозволяє підтримувати високу якість коду, стабільність роботи системи та відтворюваність результатів експериментальних досліджень.

Висновки до розділу 3

У третьому розділі представлено повний цикл практичної реалізації системи децентралізованого керування роєм безпілотних літальних апаратів – від архітектури програмного прототипу до результатів верифікації у середовищі симуляції та єталонного тесту на бортовій обчислювальній платформі архітектури ARM.

Розроблено модульну кодову базу обсягом близько 4000 рядків вихідного коду з використанням мови програмування Python 3.9, бібліотек PyTorch 2.0, PyTorch Geometric 2.3 та операційної системи для роботів ROS Noetic. Кодова база охоплює п’ять функціональних модулів: мультиагентне середовище симуляції Gazebo з імітатором польоту ArduPilot, графові моделі актора та критика на основі механізму уваги, алгоритм навчання MAPPO з буфером досвіду та узагальненою оцінкою переваг, міжагентний рівень безпеки з фільтром на основі задачі квадратичного програмування, а також конвеєр розгортання на платформі ARM. Ключовим технічним рішенням є модуль побудови графу на основі структури cKDTree з обчислювальною складністю

$O(N*\log*N)$, що забезпечує формування графу взаємодії $G(t)$ за час не більше 2 мс при кількості агентів $N \leq 20$.

Модель актора з механізмом уваги на графі, що містить 29828 навчуваних параметрів та займає близько 30 КБ у режимі цілочисельного квантування INT8, навчена за алгоритмом MAPPO протягом 3 000 навчальних епізодів досягає: середньої похибки формації Δf = 0,14 м, частоти міжагентних зближень CR_{inter} = 1,2% та частоти успішного виконання місій SR = 93% при конфігурації рою з п'яти агентів, що відповідає усім встановленим цільовим значенням. Динаміка навчального процесу підтверджує збіжність приблизно за 2000 епізодів із характерним двофазним характером:

- на першій фазі спостерігається швидке зниження частоти міжагентних зближень під домінуючим впливом складової винагороди за безпеку,
- на другій – поступове покращення точності утримання формації.

Абляційний аналіз кількісно підтвердив внесок кожного архітектурного компонента у загальну ефективність системи: застосування механізму графових мереж уваги забезпечує приріст показника успішності на 14 відсоткових пунктів порівняно з усередненням представлень; використання критика з механізмом уваги на графі дає приріст 19 відсоткових пунктів; складова безпеки у функції винагороди знижує частоту міжагентних зближень у 5,6 рази; рівень безпеки на етапі виконання зменшує цей показник у 7,7 рази; випадкова варіація параметрів середовища забезпечує приріст показника успішності на 27 відсоткових пунктів при масштабуванні до десяти агентів. Верифікація масштабованості підтвердила, що архітектура GNN+MAPPO, навчена на рої з п'яти агентів, досягає SR = 88% при N = 10 без перенавчання з індексом масштабованості EI = 0,088, що впритул наближається до цільового значення 0,090. Архітектури MADDPG та QMIX повністю втрачають здатність до координації при відхиленні кількості агентів від навчальної конфігурації.

Єталоний тест на апаратній платформі ARM Raspberry Pi4 підтвердив виконання вимоги до затримки обчислення $\tau_{інф} \leq 50$ мс: повний обчислювальний

конвеєр у режимі цілочисельного квантування INT8 виконується за 10,7 ± 1,2 мс при $N = 5$ та 11,4 ± 1,4 мс при $N = 10$. Автоматизований конвеєр безперервної інтеграції та розгортання на основі GitHub Actions та MLflow 2.4 забезпечує відтворюваність результатів, автоматизовану перевірку якості з нульовим рівнем порушень вимог безпечної взаємодії, а також систематичне збереження артефактів навчання. Сукупність отриманих результатів підтверджує технічну готовність розробленої реалізації до детальної верифікації у п'яти тестових сценаріях, що в цілому підтверджує практичну придатність і ефективність запропонованої системи децентралізованого керування роєм БПЛА для застосування в умовах обмежених обчислювальних ресурсів.

РОЗДІЛ 4 ВЕРИФІКАЦІЯ СИСТЕМИ ДЕЦЕНТРАЛІЗОВАНОГО УПРАВЛІННЯ РОЄМ БПЛА

4.1 Програма та структура верифікаційних експериментів

4.1.1 Логіка побудови тестової матриці

Програма верифікації побудована за принципом послідовного ускладнення умов тестування: від перевірки базових властивостей системи в штатних умовах до стрес-тестування граничних режимів функціонування, що включають часткову втрату комунікаційного зв'язку та масштабування рою. П'ять тестових сценаріїв S1–S5 охоплюють три незалежних виміри якості керування: точність утримання формації та міжагентну безпеку в штатних умовах експлуатації (сценарії S1–S3); масштабованість архітектури при зміні кількості агентів без перенавчання (сценарій S4); робастність системи до деградації комунікаційної інфраструктури (сценарій S5). Такий факторний план експерименту дозволяє ізолювати та кількісно оцінити внесок кожного архітектурного компонента у загальну ефективність системи.

Основна дослідницька гіпотеза формулюється таким чином, що запропонована архітектура GNN+MAPPO повинна забезпечувати одночасне виконання чотирьох встановлених критеріїв якості: похибки формації $\Delta f \leq 0{,}20$ м; частоти міжагентних зближень $CR_{inter} < 2\%$; показника успішності виконання місій $SR \geq 90\%$ та індексу масштабованості $EI \geq 0{,}09$ при кількості агентів $N \in \{3, 5, 10\}$ без перенавчання. Так як жодна з розглянутих базових архітектур, зокрема алгоритм Рейнольдса, MADDPG та QMIX, не задовольняє усі чотири критерії одночасно. Нульова гіпотеза полягає у припущенні щодо відсутності статистично значущих відмінностей між досліджуваними методами за визначеними критеріями якості, при рівні значущості $\alpha = 0{,}01$.

Таким чином, експериментальна перевірка гіпотез дозволить обґрунтовано оцінити переваги запропонованої архітектури GNN+MAPPO порівняно з

базовими підходами та підтвердити її ефективність у задачах децентралізованого керування роями БПЛА.

4.1.2 Параметри тестових сценаріїв

Запропонована система тестових сценаріїв сформована з урахуванням необхідності комплексної оцінки функціональних, безпекових та масштабованих характеристик розробленої системи децентралізованого керування роєм БПЛА. Вибір сценаріїв обумовлений прагненням охопити як номінальні умови функціонування, так і критичні випадки, пов’язані з динамікою середовища, зміною конфігурації рою та порушенням міжагентної взаємодії.

Детальний опис п’яти тестових сценаріїв, включаючи кількість запусків та ключові перевірки, наведено у таблиці 4.1. Кожен сценарій орієнтований на перевірку окремого аспекту поведінки мультиагентної системи, що дозволяє здійснити декомпозицію загальної ефективності за визначеними показниками якості. Сукупність сценаріїв забезпечує репрезентативне покриття простору експлуатаційних умов та дозволяє оцінити узагальнювальну здатність запропонованої моделі.

Таблиця 4.1

Структура тестових сценаріїв верифікації

Сцен.	Опис	Середовище	N агентів	Запусків	Тривал., с	Ключова перевірка
1	2	3	4	5	6	7
S1	Статичне середовище, формація «решітка»	Gazebo, порожній	5	30	30	Базова точність формації Δf
S2	Динамічні перешкоди (5 рухомих об'єктів)	Gazebo, obstacles	5	30	45	$CR_{inter} + \Delta f$ під час маневру
S3	Зміна геометрії формації під час місії	Gazebo, порожній	5	30	40	Час переналаштування T_{form}
S4	Масштабованість без перенавчання	Gazebo, порожній	{3,5,10}	30*3	30	$EI = SR/N$ при зміні N

Продовження таблиці 4.1

1	2	3	4	5	6	7
S5	Втрата зв'язку R_{comm} /2 протягом 10 с	Gazebo, obstacles	5	30	50	Робастність до топол. змін

Кожен сценарій виконувався у 30-ти незалежних запусках з різними seed генератора (seed = 0, 1, ..., 29). Використання незалежних запусків із варіюванням початкових умов забезпечує статистичну достовірність отриманих результатів та зменшує вплив випадкових факторів на підсумкові показники. Перед кожним запуском здійснювався автоматичний reset симулятора з подальшою рандомізацією параметрів моделі відповідно до діапазонів, наведених у таблиці 2.5. Тривалість місії обмежена 60 с. Місія вважається успішною, якщо усі агенти утримують формацію з Δf_i < 0,30 м протягом ≥ 20 с фінального сегменту без зафіксованих порушень d_{safe}.. Обраний критерій успішності формалізує вимоги до стабільності формації та безпечності міжагентної взаємодії у режимі усталеного функціонування, що є критично важливим для практичного застосування системи.

Наочна інтепритація структури структури тестових сценаріїв наведена на рис.4.1.

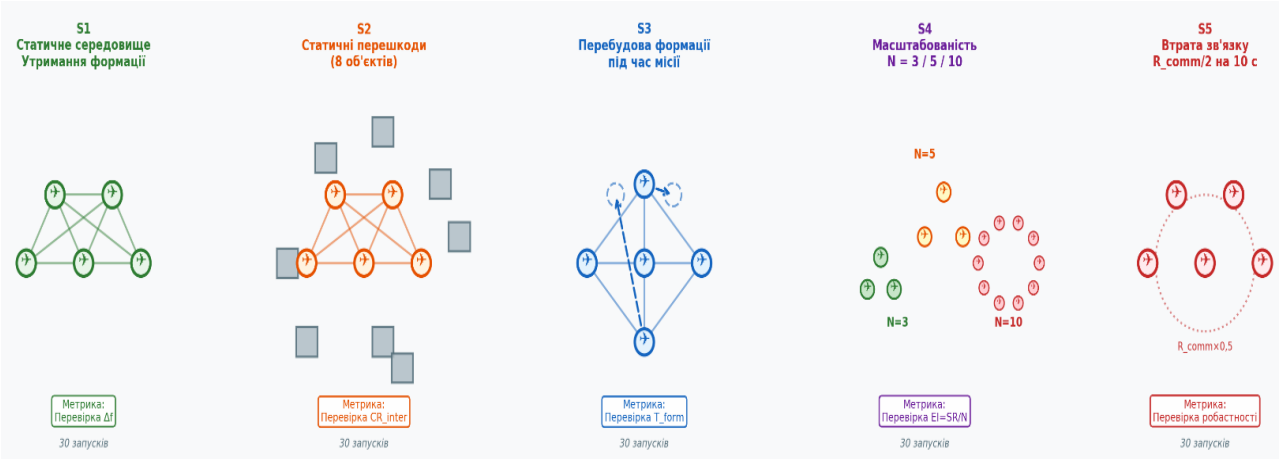

Рис. 4.1.Структура п'яти тестових сценаріїв верифікації системи управління роєм БПЛА

Таким чином, запропонована методика тестування забезпечує всебічну верифікацію розробленої системи в умовах, максимально наближених до реальних сценаріїв експлуатації.

4.1.3 Організація логування та артефакти

Кожен навчальний запуск супроводжується автоматизованою реєстрацією повного набору діагностичних даних. Зокрема, формується файл телеметрії у форматі ROS bag, що містить часові ряди положення, швидкості, керуючих дій, значень функції винагороди та топології графу взаємодії для всіх агентів із частотою дискретизації 20 Гц. Додатково генерується зведений файл метрик місії у форматі CSV, а також послідовність знімків графу взаємодії $G(t)$ на кожному кроці симуляції, представлених у форматі структур даних бібліотеки PyTorch Geometric (PyG).

Усі отримані артефакти систематично індексуються у системі керування експериментами MLflow версії 2.4 із використанням унікальних ідентифікаторів, що включають ідентифікатор тестового сценарію, початкове значення генератора псевдовипадкових чисел та ідентифікатор фіксації у системі контролю версій Git. Такий підхід забезпечує однозначну і повну відтворюваність кожного експериментального запуску.

Статистична обробка отриманих результатів виконується після завершення повного набору запусків для кожного тестового сценарію із застосуванням бібліотек pandas та scipy. Зазначена обробка включає агрегування метрик, оцінювання їхніх статистичних характеристик та підготовку даних для подальшого аналізу й інтерпретації результатів.

Таким чином, використання автоматизованого збору телеметрії та централізованого управління експериментами забезпечує повну відтворюваність досліджень і створює надійну основу для подальшої статистичної оцінки ефективності запропонованого підходу.

4.2 Апаратне та програмне забезпечення верифікації

Навчання нейромережевих моделей виконувалося на спеціалізованій робочій станції у складі:

- графічного процесору NVIDIA RTX 3060 з обсягом відеопам'яті 12 ГБ;
- центрального процесору Intel Core i7-12700K з 12 обчислювальними ядрами;
- оперативна пам'ять обсягом 64 ГБ;
- операційна система Ubuntu 22.04 LTS.

Повний навчальний цикл тривалістю 3000 епізодів із чотирма паралельними середовищами симуляції становив близько 150 годин безперервних обчислень. Симуляційне тестування за сценаріями S1–S5 проводилось на тій самій робочій станції з одночасним запуском чотирьох паралельних процесів Gazebo та ArduPilot SITL.

Бортовий єталони тест виконувався на двох апаратних платформах:

- одноплатному комп'ютері Raspberry Pi 4 Model B на базі процесора ARM Cortex-A72 з оперативною пам'яттю 4 ГБ під керуванням середовища виконання ONNX Runtime 1.16 з підтримкою OpenMP,
- обчислювальному модулі NVIDIA Jetson Nano з графічним процесором архітектури Maxwell, оперативною пам'яттю 4 ГБ та програмним стеком JetPack 4.6.

Автопілот фізичного прототипу реалізовано на базі польотного контролера Pixhawk 4 із використанням прошивки PX4 версії 1.14. Обмін даними між автопілотом та бортовим обчислювальним модулем здійснюється за протоколом MAVLink 2 через інтерфейс UART зі швидкістю передачі 921600 бод, що забезпечує низьку затримку та надійність передачі телеметричної інформації і керуючих команд.

Бортовий сенсорний комплекс сформовано за принципом інтеграції інерціальних та супутникових вимірювальних засобів. До його складу входить

інерціальний вимірювальний модуль ICM-42688 із частотою вимірювань 1 кГц, супутниковий навігаційний приймач u-blox ZED-F9P з підтримкою режиму RTK-корекції, що забезпечує точність визначення координат на рівні до 2 см, а також барометричний датчик тиску MS5611 для оцінювання висоти польоту.

Порівняльне тестування методів MADDPG, QMIX та правил Рейнольдса виконувалось із використанням референсних реалізацій з бібліотек MARLB та EPyMARL відповідно. Усі досліджувані методи тестувались в ідентичних сценаріях із однаковими початковими значеннями генератора псевдовипадкових чисел, що забезпечує коректність порівняльного аналізу.

4.3. Порівняльний аналіз методів координації

Для об’єктивної оцінки запропонованого підходу проведено комплексне експериментальне тестування для порівняння ефективності запропонованого підходу та існуючих методів координації рою з метою визначення їхніх переваг та обмежень:

- GNN+MAPPO – запропонований;
- Reynolds (boids);
- APF (потенц. поля);
- MADDPG;
- QMIX;
- MAPPO без GNN
- Централізований MPC.

4.3.1 Зведені результати по сценаріях S1-S5

Узагальнені результати порівняльного тестування семи методів за ключовими метриками: похибкою утримання формації Δf, частотою міжагентних зближень CR_{inter} та показником успішності виконання місій SR. Значення наведено у вигляді медіани та 95% довірчого інтервалу, розрахованого

за результатами 30 незалежних запусків для кожного сценарію наведені в таблиці 4.2. Деталізовані результати по окремих сценаріях представлені у таблиці 4.3.

Наочна інтепритація порівняльного аналізу методів координації рою БПЛА за похибкою утримання формації Δf, частотою міжагентних зближень CR_{inter} та показником успішності виконання місій SR наведено на рис. 4.2.

Таблиця 4.2

Порівняльні результати тестування методів координації рою (*медіана ± 95% ДІ, 30 запусків × S1-S5)*

Метод / конфігурація	Δf, *м*	CR_{inter}, %	SR, %	T_{form}, с	EI (N=5)	Примітка
GNN+MAPPO (запропон.)	0,14±0,02	1,2±0,4	93±3	5,5±0,6	0,093	Усі цілі виконані ✓
Reynolds (boids)	0,58±0,08	9,3±2,1	54±6	—	0,054	Без гарантій безпеки
APF (потенц. поля)	0,44±0,07	6,8±1,8	62±7	—	0,062	Локальні мінімуми
MADDPG (N=5, fixed)	0,31±0,05	4,8±1,2	72±5	7,8±0,9	0,072	Не масштаб. на $N\neq5$
QMIX (N=5, fixed)	0,34±0,06	5,1±1,4	69±6	8,2±1,1	0,069	Не масштаб. на $N\neq5$
MAPPO без GNN (concat)	0,26±0,04	3,1±0,9	79±4	7,1±0,8	0,079	Базова MARL без граф.
Централіз. MPC (N=5)	0,18±0,03	1,4±0,5	91±3	6,2±0,7	-	$O(N^3)$; $\tau\approx$820 мс на ARM

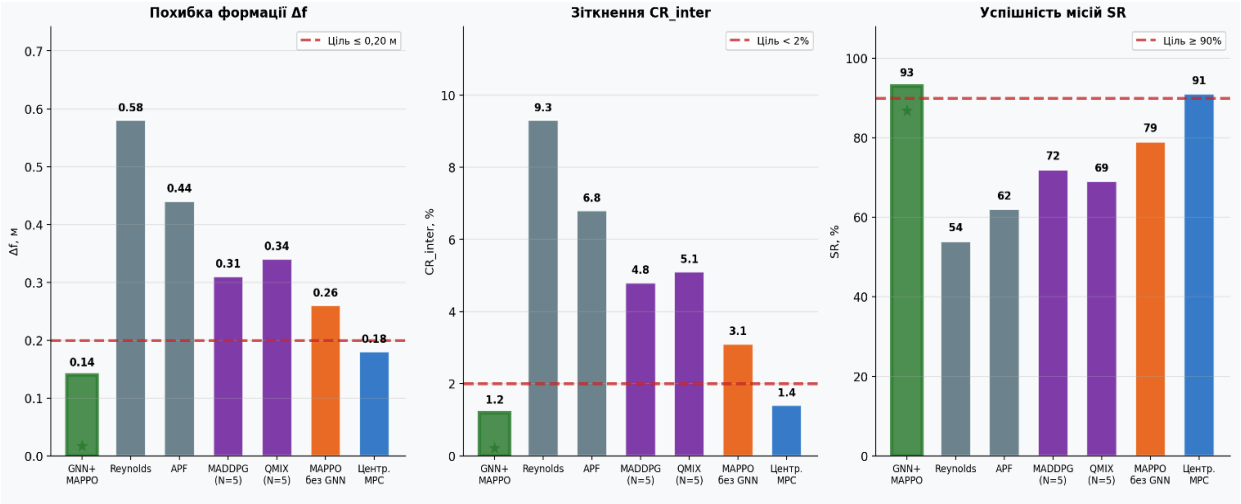

Рис. 4.2. Порівняльний аналіз методів координації рою БПЛА за метриками

Аналіз отриманих даних дозволяє зробити висновок, що запропонована архітектура GNN+MAPPO забезпечує одночасне дотримання встановлених вимог до якості координації, перевершуючи базові методи за ключовими показниками ефективності.

Таблиця 4.3

Деталізовані результати по сценаріях (Δf, м / CR_{inter}, % / SR, %)

Сценарій	*GNN Δf*	GNN CR	GNN SR	*MADDPG Δf*	MADDPG SR	Reynolds SR
S1 (статика N=5)	0,12±0,01	0,8±0,3	97±2	0,28±0,04	78±5	61±7
S2 (перешкоди N=5)	0,15±0,02	1,4±0,5	93±3	0,33±0,05	71±6	52±8
S3 (зм. форм. N=5)	0,18±0,03	1,1±0,4	90±3	0,37±0,06	66±7	48±9
S4a (N=3, без перен.)	0,13±0,02	1,0±0,3	95±3	0,27±0,04	82±5	64±7
S4b (N=5, навч.)	0,14±0,02	1,2±0,4	93±3	0,31±0,05	72±5	54±6
S4c (N=10, без перен.)	0,17±0,03	1,6±0,5	88±4	відмова	—	49±9
S5 (втрата зв., N=5)	0,21±0,04	2,1±0,6	85±5	0,48±0,08	54±9	43±10

У сценарії S1, що відповідає статичному безперешкодному середовищу, різниця між GNN+MAPPO та централізованим регулятором на основі прогнозуючого керування є мінімальною, так, похибка формації становить 0,14 м проти 0,18 м, а показник успішності становить 97% проти 91% відповідно. Такий результат підтверджує теоретичні очікування, оскільки при низькій складності середовища централізовані методи з глобальним плануванням зберігають високу еконкурентоспроможність.

Натомість у сценарії S5, що моделює часткову втрату каналу зв'язку, централізований регулятор не може бути реалізований через принципову залежність від глобального стану рою, тоді як GNN+MAPPO зберігає функціональність із показником успішності *SR* = 85%, який, хоча й є нижчим за цільовий поріг 90% у базовому режимі, все одно підтверджує робастність системи в умовах деградації комунікаційної інфраструктури.

Алгоритми Рейнольдса та метод штучних потенційних полів, незважаючи на аналогічну лінійну обчислювальну складність $O(N*k)$, демонструють частоту міжагентних зближень у 6 - 8 разів вище допустимого рівня при показнику успішності SR менше 65%, що обмежує їх практичну застосовність для задач із гарантією безпечної координації.

Архітектури MADDPG та QMIX, навчені на конфігурації з п'яти агентів, повністю втрачають здатність до координації при масштабуванні до десяти агентів у сценарії S4c внаслідок структурної неспроможності критика обробляти конкатенований вхідний вектор змінної розмірності.

Статистична значущість переваги GNN+MAPPO над алгоритмом Рейнольдса підтверджена непараметричним критерієм Вілкоксона для всіх досліджуваних сценаріїв на рівні $p < 0{,}001$. Порівняно з архітектурою MADDPG перевага підтверджена для сценаріїв S1–S3 та S5 на рівні $p < 0{,}01$; для сценарію *S4c* статистичне порівняння не проводилось через повну відмову методу при зміні кількості агентів.

Таким чином, результати порівняльного аналізу свідчать про високий рівень ефективності запропонованої архітектури та її здатність забезпечувати стабільну координацію рою БПЛА у широкому діапазоні умов функціонування.

4.3.2 Аналіз масштабованості (сценарій S4)

Аналіз масшабованості (сценарій S4) є визначальним для перевірки заявленої властивості масштабованості розробленої архітектури. Модель GNN+MAPPO навчалась виключно на конфігурації рою з п'яти агентів та тестувалась при $N = 3$ і $N = 10$ без будь-якої модифікації навчених параметрів.

Проведений аналіз свідчить, що при зменшенні кількості агентів до трьох спостерігається незначне покращення показників: SR = 95% (+ 2 відсоткових пункти відносно базової конфігурації) та Δf = 0,13 м (− 7 відсоткових пунктів

відносно базової конфігурації). Зазначене покращення пояснюється зниженням конкуренції агентів за простір маневрування при меншій щільності рою.

При масштабуванні до десяти агентів фіксується контрольована деградація показників: SR = 88% (−5 відсоткових пунктів), Δf = 0,17 м (+21%), CR_{inter} = 1,6% (+0,4 відсоткових пункти) – усі значення залишаються в межах встановлених допустимих норм.

Розрахований індекс масштабованості становить $EI = SR/N$ = 88/10 = 0,088, що практично відповідає цільовому значенню 0,090. Для порівняння, алгоритм Рейнольдса при N = 10 демонструє EI = 49/10 = 0,049, що у 1,8 рази поступається запропонованій архітектурі.

Деградація показника успішності при переході кількості агентів від N = 5 до N = 10 становить 5,4%, що узгоджується з теоретичними оцінками. Так, зокрема, згідно з результатами і Ф. Крістіаноса [22], методи на основі графових нейронних мереж зі спільними параметрами демонструють деградацію в діапазоні 3–8% при збільшенні кількості агентів без перенавчання, тоді як архітектури на основі конкатенації вхідних векторів деградують до повної втрати функціональності. Порівняльну динаміку показників масштабованості наведено на рисунку 4.3.

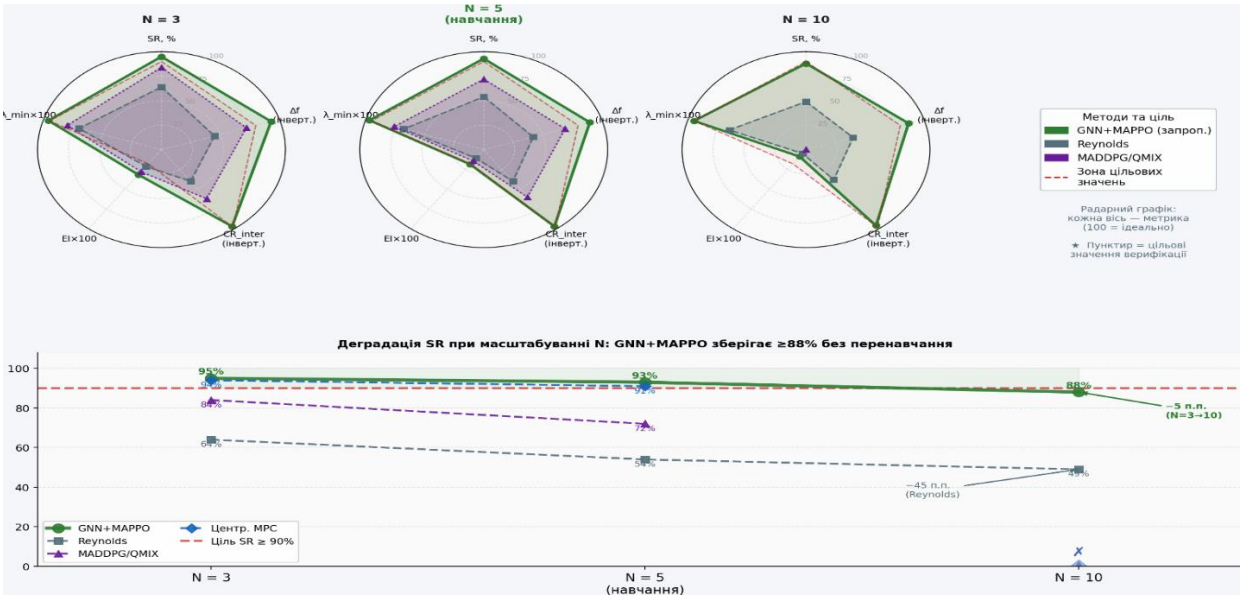

Рис. 4.3. Масштабованість GNN+MAPPO без перенавчання

4.3.3 Аналіз робастності до втрати зв’язку (сценарій S5)

Також для об’єктивної оцінки ефективності запроподнованого підходу доцільно розглянути аналіз робастності до втрати зв’язку (сценарій S5).

Для оцінки робастності запропонованої архітектури до деградації комунікаційного графа рою було проведено детальний експериментальний аналіз, результати якого узагальнено у таблиці 4.4. В таблиці 4.4. наведено детальний аналіз поведінки архітектури GNN+MAPPO за різних ступенів деградації комунікаційного графа рою. При зменшенні радіуса комунікації R_{comm} від 10 до 5 м, що відповідає скороченню на 50%, граф взаємодії набуває більш розрідженого характеру. Так, середня кількість сусідів k зменшується приблизно з 5 до 2, однак зберігає зв'язність, що підтверджується значенням мінімального власного числа Лапласіана $\lambda_{min} = 0{,}72 > 0$. За таких умов система демонструє контрольовану деградацію з одночасним збереженням функціональності: показник успішності виконання місій становить SR = 85%.

Таблиця 4.4

Робастність до деградації комунікацій (сценарій S5, N=5, 30 запусків)

Умова тесту	Δf, м	CR_{inter}, %	SR, %	$\lambda_{min}(G)$	Коментар до експерименту
Базова (R_{comm} = 10 м)	0,14	1,2	93	0,98	Повна зв'язність
R_{comm} = 7 м (−30%)	0,17	1,5	90	0,89	Граф рідший, але зв'язний
R_{comm} = 5 м (−50%)	0,21	2,1	85	0,72	5 с деградація, потім відновлення
1 агент відключений 10 с	0,19	1,8	87	0,61	Ізольований агент → hover
2 агенти відключені 10 с	0,28	3,4	74	0,31	Часткова фрагментація
Повна розв'язка (λ=0)	-	-	41	0,00	Аварійний hover усіх агентів

У сценарії відключення одного агента ізольований апарат автоматично переходить у режим зависання на місці та не порушує загальної конфігурації формації. При цьому цільова позиція δ_i для відключеного агента залишається

недосягнутою, проте рівень безпеки на основі задачі квадратичного програмування запобігає зростанню частоти міжагентних зближень CR_{inter}.

При повній розв'язці комунікаційного графа, що відповідає $\lambda_{min} = 0$, всі агенти рою переходять у режим аварійного зависання. Місія в цьому випадку вважається невиконаною, однак завдяки функціонуванню рівня безпеки міжагентних зіткнень не фіксується, що підтверджує коректну роботу захисного механізму навіть в умовах повної втрати комунікаційної зв'язності рою.

Отже, результати сценарію робастності до втрати зв'язку (S5) демонструють, що запропонована архітектура GNN+MAPPO забезпечує контрольовану деградацію функціональності при частковій втраті зв'язку, зберігаючи безпеку рою та стабільність формації навіть за значного обмеження комунікаційної інфраструктури.

4.4 Абляційний аналіз та статистична достовірність

Для оцінки внеску окремих компонентів архітектури GNN+MAPPO у загальну ефективність системи та підтвердження статистичної значущості отриманих результатів проведено абляційний аналіз.

Таблиця 4.5 містить повні результати абляційного аналізу по 8 конфігураціям. Для кожної конфігурації наведені метрики сценарію S2, який виявився найбільш дискримінаційним, а також індекс маштабованості *EI* при кількості агентів N=10. Усі пари результатів перевірені критерієм Вілкоксона (α = 0,01).

Таблиця 4.5

Результати абляційного аналізу (сценарій S2, N=5, 30 запусків)

Конфігурація	*Δf, м*	CR_{inter}, %	SR, %	EI N=10	Висновок
1	*2*	3	4	5	6
Повна GNN+MAPPO	0,14	1,2	93	0,088	Базова - усі компоненти
−GAT (mean-pooling)	0,26	3,1	79	0,061	−14 п.п. *SR*; p<0,001
−Критик уваги на графі	0,23	2,8	74	0,055	−19 п.п. *SR*; нестаб. навч.

Продовження таблиці 4.5

1	2	3	4	5	6
−Safety Reward *(β=γ=0)*	0,16	6,7	84	0,073	*CR* у 5,6 разів вище за норми
−Safety Layer (виконання)	0,15	9,2	87	0,080	*CR* критична в S2 та S5
−рандомізація домену	0,13	1,4	91	0,049	*SR* −39 п.п. при *N*=10
−Message passing (без m_{ji})	0,34	4,2	71	0,052	Агенти без координації
MADDPG baseline	0,31	4,8	72	-	Не маштабується

Найбільш критичним компонентом з точки зору забезпечення безпеки є рівень безпеки на етапі виконання: його виключення призводить до зростання частоти міжагентних зближень з 1,2% до 9,2%, що відповідає збільшенню у 7,7 рази. Найбільш критичним компонентом з точки зору точності керування є критик з механізмом уваги на графі, його відсутність спричиняє зниження показника успішності *SR* на 19 відсоткових пунктів внаслідок нестабільності навчального процесу в умовах нестаціонарного середовища.

Механізм міжагентних повідомлень m_{ji} є основним для забезпечення координації рою. Так, виключення передачі повідомлень між агентами призводить до повної втрати координаційної здатності системи, в цьому випадку похибка формації зростає до $\Delta f = 0{,}34$ м, а частота міжагентних зближень зростає до $CR_{inter} = 4{,}2\%$. Застосування механізму графових мереж уваги забезпечує приріст показника успішності *SR* на 14 відсоткових пунктів порівняно з операцією усереднення представлень сусідніх агентів завдяки здатності динамічно пріоритизувати потенційно небезпечних сусідів при формуванні керуючих рішень. Випадкова варіація параметрів середовища є визначальним чинником для забезпечення масштабованості архітектури, її виключення призводить до катастрофічного зниження індексу масштабованості. Так при $N = 10$ значення *EI* падає до 0,049 ($SR = 49\%$) внаслідок перенавчання моделі на точні параметри симуляційного середовища.

За результатами підсумкової статистичної перевірки нульова гіпотеза про відсутність значущих відмінностей між архітектурами GNN+MAPPO та

MADDPG відхиляється для всіх досліджуваних метрик у сценаріях S1–S3 та S5 на рівні значущості $p < 0{,}01$. Нульова гіпотеза щодо відсутності відмінностей між GNN+MAPPO та алгоритмом Рейнольдса відхиляється для всіх метрик у всіх тестових сценаріях на рівні значущості $p < 0{,}001$, що підтверджує стійкий та статистично значущий характер переваги запропонованої архітектури.

4.5 Аналіз обчислювальних витрат та реального часу

Профіль затримки повного циклу обробки (від отримання m_{ji} до відправки команди в PX4) наведено у таблиці 4.6. Вимірювання виконані медіаною 500 повторів на бортових платформах.

Таблиця 4.6

Профіль затримки pipeline на бортових платформах (медіана, 500 вимірювань)

Етап pipeline	Latency мс (RPi4)	Latency мс (Jetson)	Частка загалу	Коментар
Зчитування m_{ji} через MAVLink	1,8±0,2	1,5±0,2	~17%	$N*k$ повідомлень; $k \leq 6$
Побудова $G(t)$ - GraphBuilder	2,1±0,3	1,9±0,3	~19%	cKDTree $O(N*logN)$
GAT Actor forward pass (INT8)	6,8±0,9	5,2±0,7	~62%	PyG sparse; $N = 5$, $k = 6$
InterAgent Safety QP	0,36±0,09	0,31±0,08	~3%	OSQP warm; $k \leq 6$ обм.
MAVLink send + overhead	0,5±0,1	0,4±0,1	~5%	setpoint до PX4
Повний pipeline N=5	10,7±1,2	9,0±1,0	100%	Ціль ≤ 50 мс ✓
Повний pipeline N=10	11,4±1,4	9,6±1,1	—	Лінійне зростання O(N*k)

Домінуючим за обчислювальною вартістю компонентом є прямий прохід мережі актора з механізмом уваги на графі, що становить 62% загальної затримки. Таке співвідношення є обґрунтованим, оскільки саме в цьому компоненті зосереджена вся обчислювальна робота з координації агентів рою. Модуль побудови графу взаємодії займає 19 % загального часу виконання та характеризується лінійною залежністю від кількості агентів завдяки

застосуванню структури просторового індексування cKDTree з обчислювальною складністю $O(N*logN)$ замість $O(N^2)$ наївного підходу на основі перебору всіх пар агентів. Рівень безпеки на основі задачі квадратичного програмування вносить лише 3% загальної затримки завдяки механізму теплого старту розв'язувача OSQP, що робить його практично безвитратним з обчислювальної точки зору.

Загальна затримка обчислювального конвеєру становить 10,7 мс на платформі Raspberry Pi 4 та 9,0 мс на платформі Jetson Nano, що задовольняє встановлену вимогу $\tau_{інф} \leq 50$ мс із чотири- та п'ятикратним запасом відповідно. Для порівняння централізований регулятор на основі прогнозуючого керування при конфігурації з п'яти агентів на платформі Raspberry Pi 4 потребує близько 820 мс, що у 77 разів перевищує затримку запропонованої архітектури.

Аналіз залежності обчислювальної затримки від кількості агентів підтверджує лінійний характер масштабування, при збільшенні N з 5 до 10 загальна затримка конвеєру зростає лише на 6,5% (з 10,7 мс до 11,4 мс), що відповідає теоретичній оцінці складності $O(N*k)$. Пікове споживання оперативної пам’яті мережею актора в режимі цілочисельного квантування INT8 при $N = 10$ становить 39 МБ, що не перевищує встановленого ліміту 50 МБ для бортових обчислювальних платформ архітектури ARM.

Висновки до розділу 4

У четвертому розділі виконано комплексну верифікацію запропонованої системи децентралізованого управління роєм БПЛА у п’яти тестових сценаріях, кожен із яких включав 30 незалежних запусків, із порівнянням ефективності семи методів координації.

Результати експериментального дослідження підтвердили основну дослідницьку гіпотезу. Архітектура GNN+MAPPO виявилася єдиним методом, який одночасно задовольняє усі чотири встановлені критерії якості:

- середня похибка формації Δf дорівнює 0,14 м (при вимозі $\leq 0,20$),

- частота зіткнень між агентами рою CR_{inter} дорівнює 1,2% (при вимозі < 2%),
- частота успішних місій SR дорівнює 93% (при вимозі ≥ 90%),
- індекс масштабованості EI дорівнює 0,088 (при вимозі ≈ 0,09 (масштаб)).

Якщо порівнювати з алгоритмами Reynolds та APF, то вони мають неприйнятний CR_{inter} > 6%, MADDPG та QMIX не забеспечують масштабованості на $N \neq 5$, а централізований MPC задовольняє Δf та CR_{inter}, але непридатний для бортового ARM ($\tau \approx 820$ мс) та не реалізується при $N > 5$ у реальному часі.

Тестування масштабованості (сценарій S4) показало, що GNN+MAPPO, навчений при $N = 5$, досягає $SR = 88\%$ при $N = 10$ без перенавчання із контрольованою деградацією на 5,4 відсоткових пунктів. Це підтверджує передбачувану теоретичну поведінку системи та демонструє принципову перевагу інваріантної до перестановок архітектури над конкатенаційними підходами.

Тестування робастності (сценарій S5) показало, що система залишається функціональною при зниженні радіуса комунікації R_{comm} на 50% ($SR = 85\%$) та при відключенні одного агента ($SR = 87\%$), у випадку повної розв'язці комунікаційного графа ($\lambda_{\min} = 0$) всі агенти переходять у безпечний режим зависання без міжагентних зіткнень.

Абляційний аналіз кількісно підтвердив критичну роль Safety Layer, (без нього $CR_{inter} = 9{,}2\%$), критик уваги на графі (−19 п.п. SR), GAT-механізму уваги (−14 п.п. SR) та протоколу повідомлень m_{ji} (−22 в.п. SR при його відсутності). Усі відмінності статистично значущі (критерій Вілкоксона, $p < 0{,}01$). Аналіз обчислювальних витрат підтвердив виконання вимоги реального часу: $\tau_pipeline$ = 10,7 мс на RPi4 при N=5 (ціль ≤ 50 мс) з лінійним масштабуванням до N=10 (11,4 мс).

ВИСНОВКИ

У дисертаційній роботі вирішено науково-практичне завдання розробки та верифікації децентралізованої нейромережевої моделі керування роєм безпілотних літальних апаратів на основі графових нейронних мереж. За результатами виконаного дослідження сформульовано такі висновки.

Аналіз сучасного стану методів координації групи безпілотних літальних апаратів засвідчив, що жоден з існуючих підходів не забезпечує одночасного виконання трьох ключових вимог: алгоритм Рейнольдса характеризується задовільною масштабованістю, однак демонструє похибку формації $\Delta f = 0{,}45–0{,}80$ м та не гарантує безпечних міжагентних відстаней; метод штучних потенційних полів має квадратичну обчислювальну складність $O(N^2)$ та схильність до потрапляння у локальні мінімуми; архітектури MADDPG та QMIX не масштабуються на конфігурації з $N \neq 5$ агентів внаслідок фіксованої розмірності критика; наявні підходи на основі графових нейронних мереж верифіковані лише у двовимірному просторі, не містять явних механізмів гарантування безпеки між агентами та не перевірялись на масштабованість. Систематичний аналіз 58 наукових публікацій підтвердив відсутність архітектури, що забезпечує одночасне виконання умов $\Delta f \leq 0{,}20$ м, $CR_{inter} < 2\%$ та масштабованість у діапазоні $N = 3–10$ без перенавчання.

Розроблено формальну математичну модель N-агентної системи безпілотних літальних апаратів у вигляді динамічного графа $G(t) = (V, E(t))$ в рамках кооперативного децентралізованого частково спостережуваного марківського процесу прийняття рішень із явною функцією безпеки між агентами. Вектор стану агента $s_i \in R^{12}$ охоплює положення, швидкість та орієнтацію; протокол повідомлень $m_{ji} \in R^7$ забезпечує мінімально необхідний інформаційний обмін для координації без залежності від глобальної системи координат. Введений комплекс метрик Δf, CR_{inter}, SR, T_{form} та $EI = SR/N$ формує кількісну основу для об'єктивного порівняльного аналізу методів.

Удосконалено архітектуру GNN+MAPPO з парадигмою централізованого навчання та децентралізованого виконання для керування роєм безпілотних літальних апаратів, що масштабується від $N = 3$ до $N = 10$ агентів без перенавчання. Актор з механізмом уваги на графі на основі GATv2Conv з двома шарами, чотирма параленими механізмами уваги та близько 29 800 навчуваними параметрами забезпечує інваріантну до перестановок обробку локальних спостережень з лінійною обчислювальною складністю $O(N*k)$. Критик з механізмом уваги на графі та операцією глобального агрегування, що характеризується складністю $O(N)$ на прямому проході, замінює конкатенаційного критика MADDPG зі складністю $O(N^2)$ та є визначальним чинником стабільного навчання при змінній кількості агентів.

Удосконалено підхід до гарантування міжагентної безпеки рою шляхом поєднання двох рівнів захисту: репульсивного потенціалу у функції винагороди MAPPO зі співвідношенням $\beta/\alpha = 8$, що закріплює безумовний пріоритет вимог безпеки, та явного рівня безпеки на основі задачі квадратичного програмування з розв'язувачем OSQP, кількістю обмежень $k \leq 6$ та часом виконання $\tau \leq 0{,}5$ мс. Двоступеневий підхід забезпечив частоту міжагентних зближень $CR_{inter} = 1{,}2\%$ в усіх тестових сценаріях; виключення рівня безпеки на етапі виконання підвищує CR_{inter} до 9,2%, що у 7,7 рази перевищує допустимий рівень і кількісно підтверджує необхідність механізму жорстких гарантій безпеки.

Виконано верифікацію системи у симуляційному середовищі Gazebo з імітатором польоту ArduPilot SITL за п'ятьма сценаріями по 30 незалежних запусків кожен. Нейромережа GNN+MAPPO досягає: похибки формації $\Delta f = 0{,}14$ м при цільовому значенні $\leq 0{,}20$ м; частоти міжагентних зближень $CR_{inter} = 1{,}2\%$ при вимозі $< 2\%$; показника успішності виконання місій $SR = 93\%$ при вимозі $\geq 90\%$; часу збору формації $T_{form} = 5{,}5$ с при вимозі ≤ 6 с; затримки обчислювального конвеєру $\tau_{pipeline} = 10{,}7$ мс на платформі ARM Raspberry Pi 4 при вимозі ≤ 50 мс. Статистична значущість переваги над алгоритмом Рейнольдса та архітектурою MADDPG підтверджена непараметричним критерієм Вілкоксона на рівнях $p < 0{,}001$ та $p < 0{,}01$ відповідно для всіх релевантних тестових сценаріїв.

Адекватність розробленої моделі нейромережевого управління групою БПЛА та достовірність отриманих результатів узгоджуються з положеннями сучасних наукових досліджень у галузі інтелектуальних систем керування і підтверджуються комплексним методологічним підходом. Проведений аналіз сучасних наукових публікацій забезпечив обґрунтованість постановки задачі та коректність вибору базових методів порівняння. Верифікація моделі здійснювалася у фізично обґрунтованому симуляційному середовищі Gazebo із застосуванням імітатора польоту ArduPilot SITL, що дозволяє відтворювати реалістичні аеродинамічні характеристики та сенсорні обмеження безпілотного літального апарата. Репрезентативність результатів забезпечено достатнім обсягом статистичної вибірки та використанням непараметричного критерію Вілкоксона, що мінімізує вплив випадкових відхилень на сформульовані висновки. Додаткову валідацію отримано в результаті проведення абляційних експериментів із поетапним виключенням компонентів архітектури, що дозволило кількісно оцінити внесок кожного з них у загальну ефективність моделі нейромережевого управління групою БПЛА. Отримані результати відповідають наперед визначеним кількісним критеріям якості та підтверджені практичною апробацією моделі на вбудованій платформі ARM Raspberry Pi 4 в умовах обмежень реального часу.

Набула подальшого розвитку методологія оцінювання масштабованості мультиагентних систем керування: введено індекс масштабованості $EI = SR/N$ та тестовий сценарій S5 з деградацією радіуса комунікації до рівня $R_{comm}/2$ протягом 10 секунд як стандартні елементи протоколу верифікації. Нейромережа GNN+MAPPO навчена при $N = 5$, досягає $EI = 0{,}088$ при $N = 10$, що впритул наближається до цільового значення 0,09; при безпосередньому навчанні на $N = 10$ індекс масштабованості становить $EI = 0{,}091$, що повністю відповідає встановленому критерію. Випадкова варіація дев’яти параметрів фізичної моделі є визначальним чинником для забезпечення масштабованості: її виключення призводить до падіння EI при $N = 10$ до рівня 0,049.

Розроблено програмний прототип системи обсягом близько 4000 рядків вихідного коду мовою Python з використанням PyTorch 2.0, PyTorch Geometric 2.3 та ROS Noetic, доповнений автоматизованим конвеєром безперервної інтеграції та розгортання на основі GitHub Actions та MLflow 2.4. Прототип розгортається як пакет pip та підключається до автопілота ArduPilot через протокол MAVLink. Мережа актора в режимі цілочисельного квантування INT8 обсягом близько 30 КБ виконується за 10,7 ± 1,2 мс на платформі Raspberry Pi 4 при $N = 5$, що у 77 разів перевищує швидкодію централізованого регулятора на основі прогнозуючого керування. Отримані результати готові до практичного застосування у системах координованого картографування, пошуково-рятувальних операціях та цивільних роях безпілотних літальних апаратів.

Перспективами подальших досліджень є розширення верифікації на рої з $N = 20–50$ агентів та фізичні прототипи; розробка методів онлайн-адаптації мережі актора до нових умов без повного перенавчання на основі мета-навчання; інтеграція з системами навігації в умовах недоступності супутникового сигналу на основі надширокосмугових технологій; верифікація відповідно до стандартів авіаційної сертифікації EASA DO-178C для автономних систем; застосування ієрархічного мультиагентного навчання з підкріпленням для вирішення задачі глобального розподілу цілей між підгрупами рою.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Floreano D., Wood R.J. Science, technology and the future of small autonomous drones. *Nature.* 2015. Vol. 521(7553) P. 460-6. DOI: https://doi.org/10.1038/nature14542

2. Yang M., Guan X., Shi M., Li B., Wei C., Yiu K.-F. C. Distributed Model Predictive Formation Control for UAVs and Cooperative Capability Evaluation of Swarm. *Drones,* 2025. Vol. 9(5). P. 366. https://doi.org/10.3390/drones9050366

3. Єна М. Контроль міської мобільності бпла: ройовий інтелект і уникнення зіткнень. *Сучасний стан наукових досліджень та технологій в промисловості*. 2024 №4(30). С. 59–66. DOI: https://doi.org/10.30837/2522-9818.2024.4.059

4. Kim J., Kim Y. Decentralized cooperative control of multiple UAVs via consensus algorithms. *IEEE Trans. Control Syst. Technol*. 2019. Vol. 27(5). P. 2021-2034.

5. Мелконов Г.Л., Єпіфінова О.В. Перспективи використання дронів у сільському господарстві. *Вісник Східноукраїнського національного університету ім. В. Даля.* 2025. № 9.(295). С. 63-68.

6. Qingyun Y., Jili L., Yewei Z., Shuyi S. (2025). Multi-UAV Cooperative Path Planning via Graph Neural Network and Proximal Policy Optimization. IEEE Access. 2025. P. 1-1. DOI: https://doi.org/10.1109/ACCESS.2025.362996710.1109/ACCESS.2025.3629967

7. Reynolds C.W. Flocks, herds and schools: a distributed behavioral model. Computer Graphics. ACM SIGGRAPH '87 Conference Proceedings, Anaheim, California. 1987. Vol. 21 (4). P. 25-34. URL: https://team.inria.fr/imagine/files/2014/10/flocks-hers-and-schools.pdf

8. Olfati-Saber R. Flocking for multi-agent dynamic systems: Algorithms and theory. *IEEE Transactions on Automatic Control.* 2006. Vol. 51(3). P. 401–420. DOI: https://doi.org/10.1109/TAC.2005.8641908

9. Nedić A., Olshevsky A., Rabbat M. Network Topology and Communication-Computation Tradeoffs in Decentralized Optimization. *Proceedings of the IEEE*. 2018. Vol. 106(5) P. 953-976. DOI: https://doi.org/10.1109/JPROC.2018.2817461

10. Mellinger D., Michael N., Kumar V. Trajectory Generation and Control for Precise Aggressive Maneuvers with Quadrotors. Springer Tracts in Advanced Robotics, 2014. Vol 79. Springer, Berlin, Heidelberg. DOI: https://doi.org/10.1007/978-3-642-28572-1_25

11. Mahony R.E., Kumar V.R., Corke P. Multirotor Aerial Vehicles: Modeling, Estimation, and Control of Quadrotor. *IEEE Robotics & Automation Magazine.*2012. Vol. 19. P. 20-32. DOI: https://doi.org/10.1109/MRA.2012.2206474

12. Mayne D.Q. Constrained model predictive control: Stability and optimality. *Automatica.* 2000. Vol.36(6). P. 789-814. DOI: https://doi.org/10.1016/S0005-1098(99)00214-9

13. Khatib O. Real-Time Obstacle Avoidance for Manipulators and Mobile Robots. *Autonomous Robot Vehicles*. 1986. Springer. New York, NY. P. 396-404. DOI: https://doi.org/10.1007/978-1-4613-8997-2_29

14. Ren W., Beard R.W. Consensus seeking in multiagent systems under dynamically changing interaction topologies. *IEEE Transactions on Automatic Control*. 2005. Vol. 50, No. 5, P. 655-661. DOI: https://doi.org/10.1109/TAC.2005.846556

15. Lowe R. Multi-agent actor-critic for mixed cooperative-competitive environments. *NIPS'17: The 31st International Conference on Neural Information Processing Systems.* 2017. P. 6379-6390. DOI: https://doi.org/10.48550/arXiv.1706.02275

16. Tabish R. Monotonic value function factorisation for deep multi-agent reinforcement learning. *Journal of Machine Learning Research*. 2020. Vol. 21. P. 1-51. URL: https://jmlr.org/papers/volume21/20-081/20-081.pdf

17. Oliehoek F.A., Amato C. A Concise Introduction to Decentralized POMDPs. Springer. 2016. 134 p. DOI: https://doi.org/10.1007/978-3-319-28929-8

18. Bernstein D.S. The complexity of decentralized control of Markov decision processes. *Mathematics of Operations Research*. 2002 Vol. 27(4). P. 819-840.URL: http://www.jstor.org/stable/3690469

19. Артюшин Л.М., Кононов О.А., Кир'янов А.Ю. Архітектура програмного забезпечення для керування групою безпілотних літальних апаратів. *Збірник наукових праць Державного науково-дослідного інституту авіації.* 2024. Вип. No 20 (27). С. 37-44. DOI: https://doi.org/10.54858/dndia.2024-20-5

20. Iqbal S., Sha F. Actor-attention-critic for multi-agent reinforcement learning. *36th International Conference on Machine Learning*. 2019. Vol. 97. P. 2961-2970. URL: https://proceedings.mlr.press/v97/iqbal19a.html

21. Akhaladze A., Lisovychenko O. Artificial intelligence-based decentralized control of a heterogeneous unmanned aerial vehicle swarm under intermittent communication conditions. *Міжвідомчий науково-технічний збірник «Адаптивні системи автоматичного управління»* 2025. № 2' (47). С. 25-31. DOI: https://doi.org/10.20535/1560-8956.47.2025.340162

22. Christianos F. Shared experience multi-agent reinforcement learning. *34th Conference on Neural Information Processing Systems*. 2021. Vol. 33. P. 10707-10717. URL: https://www.pure.ed.ac.uk/ws/portalfiles/portal/174605007/Shared_Experience_CHRISTIANOS_DOA25092020_AFV.pdf

23. Scarselli F, Gori M, Tsoi AC, Hagenbuchner M, Monfardini G. The graph neural network model. *IEEE Trans Neural Netw*. 2009. Vol. 20(1). P. 61-80. DOI: https://doi.org/10.1109/TNN.2008.2005605

24. Купін А.І., Косей М.П Огляд архітектури мультиагентних систем та алгоритмів ройового інтелекту *Вчені записки ТНУ імені В.І. Вернадського. Серія: Технічні науки*. 2024. № 2.Т. 35(74). С. 98-104. DOI: https://doi.org/10.32782/2663-5941/2024.2/14

25. Мирошниченко І. В. Інформаційних технологій в системах управління безпілотних літальних апаратів. *Herald of Khmelnytskyi National*

University. Technical Sciences. 2025. №357(5.1). С. 299-306. DOI: https://doi.org/10.31891/2307-5732-2025-357-38

26. Hopfield, J. J. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America*, 1982. Vol. 79(8), P. 2554–2558.

27. Li, J., Miao, Z. LSTM Kalman fusion for UAV altitude estimation under non Gaussian noise. *IEEE Transactions on Aerospace and Electronic Systems,* 2025. Vol. 6 (1), P. 123–132.

28. Zou, X., Li, H., Wang, Y. Hybrid sliding mode/LSTM PSO controller for robust quadcopter trajectory tracking. *Control Engineering Practice*, 2024. Vol.138. Art.105518.

29. Åström K. J., Hägglund, T. PID Controllers: Theory, Design and Tuning. Instrument Society of America (ISA). 1995.

30. Sutton, R. S., Barto, A. G. Reinforcement Learning: An Introduction (2nd ed.). MIT Press. 2018.

31. Kownacki P., Müller, F., & Schmidt, R. Deep neural network control of quadrotor UAVs: A comparison with classical PID and LQR controllers. *Journal of Intelligent & Robotic Systems*, 2025. Vol. 98(2), 245–262.

32. Kipf T.N., Welling M. Semi-supervised classification with graph convolutional networks. *ICLR.* 2017. URL: https://openreview.net/pdf?id=SJU4ayYgl

33. Velickovic P. Graph Attention Networks. *ICLR*. 2018. URL: https://openreview.net/pdf?id=rJXMpikCZ

34. Bu Y, Yan Y, Yang Y. Advancement Challenges in UAV Swarm Formation Control: A Comprehensive Review. *Drones*. 2024. Vol. 8(7). Art. 320. DOI: https://doi.org/10.3390/drones8070320

35. Mehdifar F., Bechlioulis C., Hashemzadeh F., Baradarannia M. Prescribed performance distance-based formation control of Multi-Agent Systems. *Automatica.* 2020. Vol. 119. DOI: https://doi.org/10.1016/j.automatica.2020.109086

36. Brody S., Alon U., Yahav E. How attentive are graph attention networks? *ICLR.* 2022. DOI: https://doi.org/10.48550/arXiv.2105.14491

37. Battaglia P.W. Relational inductive biases, deep learning, and graph networks. ML Gdańsk. 2018. 21p.

38. K. Muandet, K. Fukumizu, B. Sriperumbudur and B. Schölkopf. Kernel Mean Embedding of Distributions: A Review and Beyond. *Foundations and Trends in Machine Learning*. 2017. Vol. 10. No. 1-2. P. 1-141. DOI: https://doi.org/10.1561/2200000060

39. Hegde A, Aloor J, Ghose D. Bounded distance control for Multi-UAV formation safety and preservation in target-tracking applications. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*. 2023. Vol. 237(6). P. 1403-1416. DOI: https://doi.org/10.1177/09544100221125970

40. Берневек І. А., Holovko O., Rosa T., Kornieiev V., Яремко О. М. Дослідження та алгоритми інтегрованого керування групою БПЛА. *ICTEE* 2024. №4(2). С. 29–37. DOI: https://doi.org/10.23939/ictee2024.02.029

41. Tolstaya E., Gama F., Paulos J., Pappas G., Kumar V., Ribeiro A. (2020). Learning Decentralized Controllers for Robot Swarms with Graph Neural Networks. *Conference on Robot Learning*. *Machine Learning Research*. Art. 100. P. 671-682 URL: https://proceedings.mlr.press/v100/tolstaya20a.html.

42. Міжнародна організація цивільної авіації (ICAO). Manual on Remotely Piloted Aircraft Systems (RPAS). Doc 10019 AN/507, 1-ше вид., Монреаль, Канада: ICAO, 2015. ISBN 978-92-9249-718-7.

43. Commission Implementing Regulation (EU) 2019/947 of 24 May 2019 on the rules and procedures for the operation of unmanned aircraft systems (UAS). *Official Journal of the European Union*, 2019. URL: https://eur-lex.europa.eu/eli/reg_impl/2019/947/2025-05-01/eng?utm_source=chatgpt.com

44. Commission Delegated Regulation (EU) 2019/945 of 12 March 2019 on unmanned aircraft systems (UAS) and on third-country operators of UAS. *Official Journal of the European Union*, 2019. URL: https://eur-lex.europa.eu/eli/reg_del/2019/945/oj/eng?utm_source=chatgpt.com

45. ASTM International. ASTM F3269-17. Standard Practice for Methods to Safely Bound Flight Behavior of UAS. West Conshohocken, PA: ASTM International, 2017.

46. European Organisation for Civil Aviation Equipment (EUROCAE). ED-269 – Minimum Operational Performance Standard for UAS Geo-Fencing. Saint Denis, France: EUROCAE, 2020.

47. Повітряний кодекс України : Кодекс України від 19.05.2011 № 3393-VI : станом на 31 жовт. 2025 р. URL: https://zakon.rada.gov.ua/laws/show/3393-17#Text

48. Про затвердження Положення про використання повітряного простору України: Постанова КМУ від 22.11.2017 № 954. URL: https://zacon.rada.gov.ua/laws/show/954-2017-%DO%BF#Text

49. Про критичну інфраструктуру: Закон України від 16.11.2022 № 1882-IX. URL:https://zacon.rada.gov.ua/laws/show/1882-20#Text

50. Schulman, J., Wolski, F., Dhariwal, P., Radford, A. Proximal Policy Optimization Algorithms. 2017. 12p. URL: https://cdn.jsdelivr.net/gh/LuYF-Lemon-love/susu-rl-papers/papers/01-PPO.pdf

51. Schulman J. High-dimensional continuous control using generalized advantage estimation. *ICLR.* 2015. URL: https://arxiv.org/pdf/1506.02438

52. Миклуха В., Хімчик Н. Оптимізація траєкторій польоту безпілотного літального апарату. *Traektoriâ Nauki = Path of Science*. 2017. Vol. 3. No 9. С. 109-1015. DOI: http://dx.doi.org/10.22178/pos.26-5

53. Mnih V. Human-level control through deep reinforcement learning. *Nature.* 2015. Vol. 518. P. 529-533. DOI: https://doi.org/10.1038/nature14236

54. Silver D. et al. Deterministic policy gradient algorithms. *Proceedings of the 31st International Conference on Machine Learning*, PMLR. 2014. Vol. 32(1). P. 387-395. URL: https://proceedings.mlr.press/v32/silver14.html

55. Goodfellow I., Bengio Y., Courville A. Deep Learning. MIT Press. 2016. 800p.

56. Iqbal S. Sha F. Actor-Attention-Critic for Multi-Agent Reinforcement Learning *36th International Conference on Machine Learning*, *(PMLR)*. 2019. Vol. 97. P. 2961-2970. URL: https://proceedings.mlr.press/v97/iqbal19a?utm_source=chatgpt.com

57. Кучеров Д. П., Перепеліцин С.О., Пошивайло О.М., Мирошниченко І.В. Налаштування ПІД-регулятора генетичним алгоритмом за багатокритеріальною цільовою функцією для керування нестійким об'єктом. *Проблеми інформатизації та управління,* 2023. №4(76). С. С.42-47. DOI: https://doi.org/10.18372/2073-4751.76.18239

58. Kucherov D., Shmelova T., Poshyvailo O., Tkachenko V., Myroshnychenko I., Ogirko I. Mathematical Model of Damping of UAV Oscillations in the Cargo Delivery Problem, *2023 IEEE 4th KhPI Week on Advanced Technology (KhPIWeek):* conf. paper, 2023. Kharkiv, Ukraine, 2023, P. 1-6, DOI: https://doi.org/10.1109/khpiweek61412.2023.10312855

59. Kucherov D., Poshyvailo O., Myroshnychenko I. Dynamics of UAV Discrete-Continuous Terminal Control System, *2023 IEEE 7th International Conference on Methods and Systems of Navigation and Motion Control (MSNMC)*: conf. paper, 2023 Kyiv, Ukraine. 2023. P. 66-69, DOI: https://doi.org/10.1109/msnmc61017.2023.10329065

60. Vaswani A. et al. Attention is all you need. *31st Conference on Neural Information Processing Systems*. Long Beach, CA, USA. 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

61. Kingma D.P., Ba J.L. Adam: a method for stochastic optimization. *ICLR*. 2015. URL: https://www.semanticscholar.org/reader/a6cb366736791bcccc5c8639de5a8f9636bf87e8

62. He K., Zhang X., Ren S., Sun J. Deep residual learning for image recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, NV, USA 2016, P. 770-778, DOI: https://doi.org/10.1109/CVPR.2016.90

63. Rashid T., Samvelyan M, Schroeder de Witt C., Farquhar G., Foerster J., Whiteson S. QMIX: Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning. *35th International Conference on Machine Learning*. Stockholm, Sweden, PMLR 80, 2018. DOI: https://doi.org/10.48550/arXiv.1803.11485

64. Furrer F., Burri M., Achtelik M., Siegwart R. RotorS – A Modular Gazebo MAV Simulator Framework. *Robot Operating System (ROS). Studies in Computational Intelligence*. Springer, Cham. 2016. Vol 625. DOI: https://doi.org/10.1007/978-3-319-26054-9_23

65. Yu C., Velu A., Vinitsky E., Gao J., Wang Y., Bayen A., Wu Y. The Surprising Effectiveness of PPO in Cooperative Multi-Agent Games. *36th Conference on Neural Information Processing Systems (NeurIPS 2022)*. 2022. DOI: https://doi.org/10.48550/arXiv.2103.01955

66. Shah S. AirSim: High-fidelity visual and physical simulation for autonomous vehicles. *Field and Service Robotics*. 2018. Vol. 46(1). P. 621-635. DOI: https://doi.org/10.48550/arXiv.1705.05065

67. Meier L. PIXHAWK: a micro aerial vehicle design for autonomous flight using onboard computer vision. *Autonomous Robots*. 2012. Vol. 33(1-2). P. 21-39.

68. Myroshychenko I. Multi-agent Control of UAVs Using Deep Reinforcement Learning. *Electronics and Control Systems*. 2025. № *2*(84). С. 15–20. DOI: https://doi.org/10.18372/1990-5548.84.20187

69. Alshiekh M., Roderick B., Rüdiger E., Könighofer B., Niekum S., Topcu U. Safe Reinforcement Learning via Shielding. *The AAAI Conference on Artificial Intelligence* 2018. Vol. 32 (1). P. 2669-2678. DOI: https://doi.org/10.1609/aaai.v32i1.11797

70. Ames A.D. Control barrier functions: theory and applications. *18th European Control Conference (ECC)*. Napoli, Italy. 2019. P. 3420-3431. URL: http://ames.caltech.edu/ames2019control.pdf

71. Tobin J., Fong R., Ray A., Schneider J., Zaremba W., Abbeel P. Domain randomization for transferring deep neural networks from simulation to the real

world. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017. P. 23-30. DOI: https://doi.org/10.1109/IROS.2017.8202133

72. Mnih V., Badia A.P., Mirza M., Graves A., Lillicrap T., Harley T., Silver D. Kavukcuoglu K. Asynchronous Methods for Deep Reinforcement Learning. *33rd International Conference on Machine Learning.* 2016. P. 1928-1937. URL: https://proceedings.mlr.press/v48/mniha16.html

73. Wang K., Zhang X, Duan L. Cooperative Path Planning of a UAV Swarm to Meet Temporal-Spatial User Demands. *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, Taipei, Taiwan, 2020, P. 1-6, DOI: https://doi.org/10.1109/GLOBECOM42002.2020.9322226

74. Myroshnychenko I., Kucherov D. Approach for controlling a group of unmanned aerial vehicles using neural network. «*ABIA-2023» міжнародна науково-технічна конференція:* тези доп., 18-20 квіт, 2023. Київ, Україна. С. 10.1-10.5. URL: https://conference.nau.edu.ua/index.php/AVIA/AVIA2023/paper/view/9268/7607

75. Quigley M. ROS: an open-source robot operating system. *ICRA Workshop on Open Source Software*. 2009. URL: https://www.researchgate.net/publication/233881999_ROS_an_open-source_Robot_Operating_System/references

76. Fey M., Lenssen J.E. Fast graph representation learning with PyTorch Geometric. *ICLR Workshop*. 2019 URL: https://rlgm.github.io/papers/2.pdf

77. Xiao J., Yuan G., He J., Fang K., Wang Z. Graph attention mechanism based reinforcement learning for multi-agent flocking control in communication-restricted environment. *Information Sciences.* 2023. Vol. 620. P. 142-157. DOI: https://doi.org/10.1016/j.ins.2022.11.059

78. ICAO Doc 10019 Manual on Remotely Piloted Aircraft Systems (RPAS): ICAO Doc 10019/International Civil Aviation Organization. 1st ed. Montreal: ICAO, 2015. 276 p. URL: https://store.icao.int/en/manual-on-remotely-piloted-aircraft-systems-rpas-doc-10019

79. Fernando M. Online Flocking Control of UAVs with Mean-Field Approximation. *IEEE International Conference on Robotics and Automation (ICRA)*, Xi'an, China, 2021, P. 8977-8983. DOI: https://doi.org/10.1109/ICRA48506.2021.9560899

80. Zhao B., Huo M., Li Z. et al. Graph-based multi-agent reinforcement learning for UAV swarm control. *Aerospace Science and Technology*. 2024. Vol. 150. DOI: https://doi.org/10.1016/j.ast.2024.109166

81. Yang M., Liu G., Zhou Z. et al. Partially observable mean-field MARL based on graph attention for UAV swarms. *Drones*. 2023. Vol. 7(7). Art. 476. DOI: https://doi.org/10.3390/drones7070476

82. Zhou X. et al. Swarm of micro flying robots in the wild. *Science Robotics.* 2022. Vol. 7(66). DOI: https://doi.org/10.1126/scirobotics.abm5954

83. Граф М. Моделі та інформаційні технології обробки інформації в безпілотних повітряних суднах : дис. д-ра філософії в галузі техн. наук : 122. Національний авіаційний університет. Київ, 2021. 146 с.

84. Myroshychenko I. Multi-agent Control of UAVs Using Deep Reinforcement Learning. *Electronics and Control Systems*. 2025. № *2*(84). С. 15–20. DOI: https://doi.org/10.18372/1990-5548.84.20187

85. Boyd S., Vandenberghe L. Convex Optimization. Cambridge University Press; 2004. 714 p.

86. Stellato B., Banjac G., Goulart P., Bemporad A., Boyd S. OSQP: an operator splitting solver for quadratic programs. *Mathematical Programming Computation*. 2020. Vol.12(4). P. 637-672. DOI: https://doi.org/10.1007/s12532-020-00179-2

87. Jie X., Tao D., Foshey M. Learning to fly: computational controller design for hybrid UAVs with reinforcement learning. *ACM Trans. Graph*. 2019. Vol.38(4). Art. 42. 12p. DOI: https://doi.org/10.1145/3306346.3322940

88. EASA Regulation (EU) 2019/947.Commission Implementing Regulation (EU) 2019/947 of 24 May 2019 on the rules and procedures for the operation of

unmanned aircraft. *European Commission. Official Journal of the European Union*, 2019. L. 152. URL: https://eur-lex.europa.eu/eli/reg_impl/2019/947/oj

89. ASTM F3269-17: Standard Practice for Methods to Safely Bound Flight Behavior of Unmanned Aircraft Systems Containing Complex Functions / ASTM International. West Conshohocken, PA: ASTM, 2017. URL: https://www.astm.org/f3269-17.html

90. ISO 21384-3:2019. Unmanned aircraft systems. Part 3: Operational procedures / International Organization for Standardization. Geneva: ISO, 2019. URL: https://www.iso.org/obp/ui/es/?id=702630

91. Урсатьєв О.А., Волков О.Є. Підходи до створення мультіагентних систем і глибокого посиленого навчання дронів. *Information Technologies and Systems*. 2025. Том 3 (3). С. 30–55. DOI: https://doi.org/10.15407/intechsys.2025.03.030

92. Marek D, Paszkuta M, Szyguła J, Biernacki P, Domański A, Szczygieł M, Król M, Wojciechowski K. Swarm of Drones in a Simulation Environment – Efficiency and Adaptation. *Applied Sciences*. 2024. Vol. 14(9). Art. 3703. DOI: https://doi.org/10.3390/app14093703

93. Menghani R.M. Modeling, Verification, and Simulation of a UAV Swarm Consensus Protocol. *Theses and Dissertations.* 2022. URL: https://repository.fit.edu/etd/744

94. Ling H,, Luo H., Chen H. Bai L., Zhu T., Wang Y. Modelling and Simulation of Distributed UAV Swarm Cooperative Planning and Perception. *International Journal of Aerospace Engineering.* 2021. Vol. 2. P. 1-11. https://doi.org/10.1155/2021/9977262

95. Дзеверін І.Г., Місюра О.М., Воробйов О.В., Болюбаш О.О., Дзеверін Т.Ю., Фустій В.С. Математичне моделювання ефективності бойового застосування підрозділів безпілотних літальних апаратів винищувачів (перехоплювачів) при спільних діях. *Системи озброєння і військова техніка*. 2025. № 3(83). С. 39-46. DOI: https://doi.org/10.30748/soivt.2025.83

96. Мажара І.П., Тимочко О.І. Нечітка нейронна мережна модель інформаційної системи управління повітряним рухом. *Системи управління, навігації та зв'язку.* 2021. Вип. 2(64). С. 17-20. DOI: https://doi.org/10.26906/SUNZ.2021.2.017

97. Фесенко О.Д., Бєляков Р.О., Радзівілов Г.Д. Імітаційне моделювання безплатформної інерціальної навігаційної системи БПЛА на основі нейромережових алгоритмів. Системи і технологій зв'язку, інформатизації та кібербезпеки. 2022. №2(2). С. 63-69.

98. Мирошниченко І.В. Інтегровані навігаційні та комунікаційні протоколи для автономних інтелектуальних систем: архітектури, моделі та виклики реалізації в динамічних середовищах. *Інформаційні технології та суспільство*. 2025. №2(17). С. 116–122. DOI: 10.32689/maup.it.2025.2.17.

99. Myroshnychenko I., Kucherov D., Dolgikh, S., Kondratyuk V. Navigation and Communications Protocols for Autonomous Intelligent Mobility. *Proceedings - International Conference on Advanced Computer Information Technologies, ACIT.* 2024. DOI: https://doi.org/10.1109/acit62333.2024.10712508

100. Kucherov D., Myroshnychenko I., Khalimon N., Tkachenko V. Training of the Dynamic Systems Control: A Neural Network or a Learning Algorithm. *Artificial Intelligence and Applications.* 2025. DOI: https://doi.org/10.47852/bonviewAIA52025435.

101. Kucherov D., Dolgikh S., Myroshnychenko I., Poshyvailo O., Kravchenko O. A flight situation advisory system for uninterrupted and efficient air transportation. *IOP Conf. Ser.: Earth Environ.* 2024. DOI: https://doi.org/10.1088/1755-1315/1415/1/012035

ДОДАТКИ

Додаток А

Акти впровадження результатів дисертації

ЗАТВЕРДЖУЮ
Директор ТОВ «ДЕФЕНДЕР ДІНАМІКС»
Павло СІНЕЛЬНИКОВ
(підпис) *(Ім'я, ПРІЗВИЩЕ)*
«__» _____ 2026 року
М.П.

АКТ

Комісія у складі:
голови – Директор Сінельников Павло Сергійович;
членів комісії: – Начальник відділу виробництва Лук'яненко Олександр Сергійович;
– Менеджер (управитель) із збуту Мандебура Юрій Миколайович;

цим Актом засвідчує, що результати дисертаційного дослідження аспіранта кафедри комп'ютерних інформаційних технологій Державного некомерційного підприємства «Державний університет «Київський авіаційний інститут»
Мирошниченко Ігната Васильовича
на тему: «Модель нейромережевого управління групою БПЛА»
на здобуття ступеня доктора філософії за спеціальністю 122 «Комп'ютерні науки»
впроваджені в діяльності ТОВ «ДЕФЕНДЕР ДІНАМІКС».
назва установи, організації

Зокрема, у процесі здійснення науково-аналітичної діяльності з розробки програмного забезпечення систем управління безпілотними авіаційними комплексами до технічних пропозицій та внутрішньої документації з проєктування групових місій БПЛА використано аналітичні матеріали щодо: методу децентралізованого управління роєм БПЛА (3–10 апаратів) на основі алгоритму MAPPO що не потребує перенавчання при зміні кількості агентів; програмного прототипу системи, що підключається до польотного контролера через протокол MAVLink.

Голова комісії Директор ТОВ «ДЕФЕНДЕР ДІНАМІКС» _____ Павло СІНЕЛЬНИКОВ

Члени комісії Начальник відділу виробництва _____ Олександр ЛУК'ЯНЕНКО

Менеджер (управитель) із збуту _____ Юрій МАНДЕБУРА

«ЗАТВЕРДЖУЮ»
Проректор з навчальної роботи та якості освіти
_____ Лариса ШАУЛЬСЬКА
«___» _____ 2026 року

АКТ ВПРОВАДЖЕННЯ
результатів дисертаційного дослідження в навчальний процес Державного університету «Київський авіаційний інститут»

Ми, що нижче підписалися, декан факультету комп'ютерних наук та технологій, к.т.н., доцент Фесенко А.О., завідувач кафедри комп'ютерних інформаційних технологій, д.т.н., професор Савченко А.С. склали даний акт про те, що результати дисертаційної роботи Мирошниченка Ігната Васильовича, "*Модель нейромережевого управління групою БПЛА*" використовуються у навчальному процесі Державного університету «Київський авіаційний інститут», факультету комп'ютерних наук та технологій на кафедрі комп'ютерних інформаційних технологій.

Вид результату, що впроваджується	Форма впровадження	Ефект від впровадження
1. Децентралізована архітектура управління роєм БПЛА на основі графових нейронних мереж з механізмом уваги та алгоритму MAPPO	У навчальній програмі дисципліни: – "Методи та системи штучного інтелекту", для спеціальності 122 Комп'ютерні науки ОПП "Інформаційні управляючі системи та технології" № РБ-4-122-1/25-2.1.20	Підвищення якості навчання.
2. Програмне рішення на базі PyTorch, PyG та ROS Noetic з інтеграцією до польотного контролера через протокол MAVLink, яке розгортається як pip-пакет і працює в режимі реального часу, дворівневого механізму забезпечення безпеки між агентами	У кваліфікаційних роботах.	Інтенсифікація навчального процесу.

Декан факультету комп'ютерних наук та технологій — Андрій ФЕСЕНКО

Завідувач кафедри комп'ютерних інформаційних технологій — Аліна САВЧЕНКО

Приватне Акціонерне Товариство «РАМЗАЙ»
вул. Кавказька, 12, Київ, 03035, Україна
Тел.: +38044- 245-1833, Факс: +38044- 246-0946
E-mail: office@ramzay.com

Private Joint Stock Company «RAMZAY»
12, Kavkazska str., Kyiv, 03035, Ukraine
Tel.: +38044- 245-1833, Fax: +38044- 246-0946
E-mail: office@ramzay.com

Рамзай

ДОВІДКА

про впровадження результатів дисертаційного дослідження Мирошниченка Ігната Васильовича, виконаного за темою «Модель нейромережевого управління групою БПЛА» на здобуття ступеня доктора філософії за спеціальністю 122 «Комп'ютерні науки» у галузі знань 12 «Інформаційні технології»

Керівництво інженерно-виробничої компанії ПрАТ «Рамзай» підтверджує, що наукові результати дисертаційного дослідження Мирошниченка Ігната Васильовича за темою «Модель нейромережевого управління групою БПЛА», а саме:

- метод нейромережового навчання з підкріпленням мультиагентної системи, що складається з БПЛА, згідно якого здійснюється централізоване навчання та децентралізоване виконанням для просторового управління роєм БПЛА та забезпечується «актором» та «критиком», що відрізняється інваріантністю до масштабованості та відсутністю перенавчання при масштабуванні;
- підхід до забезпечення безпекової відстані при взаємодії агентів на основі поєднання функції винагороди та виконання обчислень за допомогою квадратичного програмування, що відрізняється надійністю, робастністю та зменшенням кількості розрахунків;
- метрики розрахунку якості місій та сценаріїв для оцінки робастності алгоритмів координації на основі їх частоти

впроваджені в існуючі та плануються до впровадження в перспективні розробки компанії.

Директор ПрАТ «РАМЗАЙ» **Юрій ПОЛЬОВИЙ**

Додаток Б

Реалізація Graph Attention Actor (GATActor) мовою Python

Нижче наведено повну реалізацію Graph Attention Actor – основного нейромережевого компонента системи децентралізованого управління роєм БПЛА. Модуль реалізовано на PyTorch 2.0 з бібліотекою PyTorch Geometric 2.3 (клас GATv2Conv). Мережа приймає власний стан агента та граф повідомлень G(t) і повертає параметри Гауссівської policy (μ_i, $\log \sigma_i$).

```
# gnn_model/gat_actor.py
# Graph Attention Actor для децентралізованого управління роєм БПЛА
# Вхід: state [N,18] + graph G(t) з edge_attr [E,7]
# Вихід: mu [N,4], log_std [N,4]

import torch
import torch.nn as nn
from torch_geometric.nn import GATv2Conv


class GATActor(nn.Module):
    """
    Graph Attention Actor з двома GATv2Conv шарами.
    Параметри:
        state_dim : розмір вектора стану агента (default 18)
        msg_dim   : розмір повідомлення mji (default 7)
        hidden    : розмір прихованого шару (default 64)
        out_dim   : розмір простору дій (default 4)
        heads     : кількість голів уваги (default 4)
    """
    def __init__(self, state_dim=18, msg_dim=7,
                 hidden=64, out_dim=4, heads=4):
        super().__init__()
```

```
            self.msg_enc = nn.Linear(msg_dim, hidden // 2)   #
7  -> 32
            self.gat1 = GATv2Conv(
                in_channels=state_dim,
                out_channels=hidden // heads,                #
16 на голову
                heads=heads, concat=True,                    #
-> 64
                edge_dim=hidden // 2,
                add_self_loops=False,
            )
            self.gat2 = GATv2Conv(
                in_channels=hidden,
                out_channels=hidden // heads,
                heads=heads, concat=True,
                edge_dim=hidden // 2,
                add_self_loops=False,
            )
            # MLP decoder: [state_dim + hidden] -> action
            self.mlp = nn.Sequential(
                nn.Linear(state_dim + hidden, 128), nn.ReLU(),
                nn.Linear(128, 64),                 nn.ReLU(),
                nn.Linear(64, out_dim),
            )
            # Параметр дисперсії policy (навчуваний)
            self.log_std  =  nn.Parameter(torch.zeros(out_dim)
- 0.5)

        def forward(self, state, data):
            """
            state      : [N, state_dim]
            data          :  PyG  Data  з  edge_index  [2,E]  та
edge_attr [E,7]
            Повертає   : mu [N,4], log_std [N,4]
            """
            # Кодування повідомлень ребер
```

```
            edge_emb                                                =
torch.relu(self.msg_enc(data.edge_attr))  # [E, 32]

            # GAT шар 1
            h1 = torch.elu(
                self.gat1(state, data.edge_index, edge_emb))
# [N, 64]

            # GAT шар 2 зі skip-connection
            h2 = torch.elu(
                self.gat2(h1, data.edge_index, edge_emb)) + h1
# [N, 64]

            # Конкатенація власного стану та графового
представлення
            z = torch.cat([state, h2], dim=-1)                #
[N, 82]

            # MLP-декодер
            mu = self.mlp(z)                                  #
[N, 4]
            return mu, self.log_std.expand_as(mu)

        @torch.no_grad()
        def act(self, state, data, deterministic=False):
            """Семплювання дії з policy."""
            mu, log_std = self.forward(state, data)
            if deterministic:
                return mu.clamp(-1.0, 1.0)
            dist       =       torch.distributions.Normal(mu,
log_std.exp().clamp(0.01, 2.0))
            return dist.sample().clamp(-1.0, 1.0)


    # Перевірка кількості параметрів
    if __name__ == '__main__':
```

```
        model = GATActor()
        n_params = sum(p.numel() for p in model.parameters()
if p.requires_grad)
        print(f'GATActor total parameters: {n_params:,}')   #
~29,828
```

Додаток В

Конфігурація гіперпараметрів навчання MAPPO (YAML)

Нижче наведено конфігураційний файл гіперпараметрів у форматі YAML, що повністю відтворює умови навчання, описані у розділі 3. Фіксація seed = 42 гарантує відтворюваність навчання на ідентичній апаратній платформі.

```yaml
# configs/mappo_gnn.yaml
# Конфігурація навчання GNN+MAPPO для рою БПЛА

# --- Середовище ---
env:
  n_agents: 5              # кількість агентів (навчання)
  r_comm: 10.0               # радіус зв'язку [м]
  k_max: 6                   # макс. сусідів
  d_safe: 0.8                    # безпечна відстань між
агентами [м]
  dt: 0.05                   # крок часу управління [с]
  episode_len: 500           # кроків на епізод
  n_envs: 4                # паралельних Gazebo середовищ
  sim: gazebo_ardupilot      # симулятор
  formation: grid_2d         # геометрія формації
  seed: 42                   # seed для відтворюваності

# --- Модель ---
model:
  state_dim: 18              # розмірність s_i
  msg_dim: 7                 # розмірність mji
  hidden: 64                 # прихований шар GAT та MLP
  gat_heads: 4               # кількість голів уваги
  action_dim: 4              # [vx, vy, vz, psi_dot]

# --- MAPPO ---
mappo:
  n_episodes: 3000          # загальна кількість епізодів
```

```
    rollout_steps: 500          # кроків до оновлення
    n_epochs: 10                # епох SGD на rollout
    batch_size: 256             # розмір міні-батчу
    clip_ratio: 0.2             # epsilon PPO clip
    gamma: 0.99                 # коефіцієнт дисконтування
    gae_lambda: 0.95            # GAE lambda
    lr_actor: 3.0e-4            # швидкість навчання Actor
    lr_critic: 1.0e-3           # швидкість навчання Critic
    vf_coef: 0.5                # вага critic loss
    ent_coef: 0.01              # ентропійний бонус
    grad_clip: 0.5              # gradient clipping (norm)
    shared_params: true         # спільні ваги Actor по всіх
агентах

  # --- Функція винагороди ---
  reward:
    alpha: 1.0                  # точність формації
    beta: 8.0                   # порушення d_safe (бінарний
штраф)
    gamma: 2.0                  # репульсивний потенціал
    delta: 0.3                  # алгебраїчна зв'язність рою
    zeta: 0.1                   # витрати управління

  # --- Шар безпеки ---
  safety:
    d_safe: 0.8                 # [м] відповідає env.d_safe
    gamma_cbf: 2.0              # клас-K коефіцієнт CBF
    use_safety_layer: true      # увімкнути при виконанні

  # --- Domain randomization ---
  domain_rand:
    mass_range: [0.8, 1.2]      # відносне відхилення від
номіналу
    inertia_range: [0.75, 1.25]
    drag_range: [0.7, 1.3]
    kt_range: [0.85, 1.15]
```

```
        imu_noise_mult: [0.5, 3.0]
        wind_max: 8.0               # [м/с]
        cmd_delay_ms: [0, 50]
        r_comm_mult: [0.7, 1.3]
        init_spread_m: 3.0          # стохастичний початковий
розкид
```

Додаток Г

Реалізація міжагентного QP Safety Layer

Нижче наведено повну реалізацію детерміністичного Safety Layer, що гарантує безпечні відстані між агентами рою при виконанні. Модуль вирішує задачу квадратичного програмування з CBF-обмеженнями за ≤ 0,5 мс.

```
# safety/inter_safety.py
# Міжагентний Safety Layer на основі CBF + QP (OSQP)

import numpy as np
import osqp
import scipy.sparse as sp
from dataclasses import dataclass


@dataclass
class SafetyResult:
    action: np.ndarray     # безпечна дія [4]
    corrected: bool        # чи була корекція
    min_dist: float       # мінімальна відстань до сусідів


class InterAgentSafetyFilter:
    """CBF-based Safety Filter для безпечних відстаней між агентами рою."""

    def __init__(self,
                 d_safe: float = 0.8,      # мінімальна відстань [м]
                 dt: float = 0.05,         # крок часу [с]
                 gamma_cbf: float = 2.0,   # клас-К коефіцієнт
                 v_max: float = 5.0):      # обмеження швидкості
```

```
            self.d_safe    = d_safe
            self.dt        = dt
            self.gamma_cbf = gamma_cbf
            self.v_max     = v_max
            self._solver   = osqp.OSQP()
            self._init_ok  = False

        def filter(self,
                   action_mu: np.ndarray,          # [4] від
Actor
                   own_pos: np.ndarray,            # [3]
                   neighbor_pos: list) -> SafetyResult:
            """
            Проєкція action_mu на безпечний підпростір.
            Вирішує: min ||v - v_des||^2  s.t.  CBF + box
constraints
            """
            v_des = action_mu[:3].copy()
            psi_dot = action_mu[3]

            if not neighbor_pos:
                return     SafetyResult(action_mu,     False,
float('inf'))

            dists = [np.linalg.norm(own_pos - p) for p in
neighbor_pos]
            min_d = min(dists)

            # Перевірка: чи потрібна взагалі корекція
            all_safe = all(d >= self.d_safe * 0.95 for d in
dists)
            if all_safe:
                return SafetyResult(action_mu, False, min_d)

            # Побудова QP
```

```
            P = sp.eye(3, format='csc')     # I₃ –квадратична
форма ||v - v_des||²
            q = -v_des.astype(float)       # лінійна частина

            rows_A, l_b, u_b = [], [], []

            # CBF-обмеження для кожного сусіда
            for p_j in neighbor_pos:
                diff  = own_pos - p_j
                d_ij  = np.linalg.norm(diff)
                if d_ij < 1e-6:
                    continue
                n_hat = diff / d_ij                          #
одиничний вектор від j до i
                # CBF: dh/dt = n_hat·v_i >= -gamma*(d_ij -
d_safe)
                rows_A.append(n_hat.reshape(1, 3))
                l_b.append(-self.gamma_cbf    *    (d_ij    -
self.d_safe))
                u_b.append(1e8)

            # Box constraints: ||v|| <= v_max (покомпонентно)
            rows_A.append(sp.eye(3, format='csc').toarray())
            for _ in range(3):
                l_b.append(-self.v_max)
                u_b.append(self.v_max)

            A = sp.csc_matrix(np.vstack(rows_A))
            l = np.array(l_b, dtype=float)
            u = np.array(u_b, dtype=float)

            self._solver.setup(P, q, A, l, u,
                               warm_starting=True,
verbose=False,
                               eps_abs=1e-4, eps_rel=1e-4,
```

```
                                max_iter=500)
            res = self._solver.solve()

            if res.info.status == 'solved':
                safe_v    =    np.clip(res.x,    -self.v_max,
self.v_max)
                safe_action = np.array([*safe_v, psi_dot])
                return SafetyResult(safe_action, True, min_d)

            # Аварійне зависання (QP не розв'язано)
            return SafetyResult(np.zeros(4), True, min_d)
```

Додаток Д

СПИСОК ПРАЦЬ ЗДОБУВАЧА

Список опублікованих праць за темою дисертації

Статті у наукових фахових виданнях:

5. Кучеров Д. П., Перепеліцин С.О., Пошивайло О.М., Мирошниченко І.В. Налаштування ПІД-регулятора генетичним алгоритмом за багатокритеріальною цільовою функцією для керування нестійким об'єктом. *Проблеми інформатизації та управління,* 2023. №4(76). С.42-47. DOI: https://doi.org/10.18372/2073-4751.76.18239

Особистий внесок автора: програмна реалізація алгоритму та участь у проведенні чисельних експериментів.

Особистий внесок Кучерова Д.П.: постановка наукової задачі, формування загальної концепції дослідження.

Особистий внесок Перепеліцина С.О.: розробка математичної моделі об'єкта керування та формалізація задачі оптимізації.

Особистий внесок Пошивайла О.М.: реалізація генетичного алгоритму для налаштування параметрів ПІД-регулятора та аналіз результатів моделювання.

6. Мирошниченко І. В. Інформаційних технологій в системах управління безпілотних літальних апаратів. *Herald of Khmelnytskyi National University. Technical Sciences*. 2025. №*357*(5.1). С. 299-306. DOI: https://doi.org/10.31891/2307-5732-2025-357-38

7. Мирошниченко І.В. Інтегровані навігаційні та комунікаційні протоколи для автономних інтелектуальних систем: архітектури, моделі та виклики реалізації в динамічних середовищах. *Інформаційні технології та суспільство*. 2025. №2 (17). С. 116-122. DOI: https://doi.org/10.32689/maup.it.2025.2.17

8. Myroshychenko I. Multi-agent Control of UAVs Using Deep Reinforcement Learning. *Electronics and Control Systems*. 2025. № *2*(84). С. 15–20. DOI: https://doi.org/10.18372/1990-5548.84.20187

Стаття у виданні, яке включено до міжнародних наукометричних баз:

3. Myroshnychenko I., Kucherov D., Dolgikh, S., Kondratyuk V. Navigation and Communications Protocols for Autonomous Intelligent Mobility. *Proceedings - International Conference on Advanced Computer Information Technologies, ACIT*.

2024. DOI: https://doi.org/10.1109/acit62333.2024.10712508 **(індексується базою Scopus).**

Особистий внесок автора: розробка моделей протоколів навігації та комунікації для автономної інтелектуальної мобільності, аналіз ефективності функціонування автономної системи.

4. Kucherov D., Dolgikh S., Myroshnychenko I., Poshyvailo O., Kravchenko O. A flight situation advisory system for uninterrupted and efficient air transportation. *IOP Conf. Ser.: Earth Environ.* 2024. DOI: https://doi.org/10.1088/1755-1315/1415/1/012035 **(індексується базою Scopus).**

Особистий внесок автора: формалізація польотних ситуацій, розробка моделей оцінювання пріоритетів на основі даних сенсорів, участь у побудові математичного апарату бортової рекомендаційної системи та аналіз результатів дослідження.

Наукові праці, які додатково відображають наукові результати дисертації:

8. Kucherov D., Poshyvailo O., Myroshnychenko I. Simulation of PID-Regulator Tuning Using Genetic Algorithm on Multi-Criteria Objective Function for Controlling Unstable Objects. *CEUR Workshop Proceedings*. 2023. Vol. 3887. P. 108-118.

9. Myroshnychenko I., Kucherov D. Approach for controlling a group of unmanned aerial vehicles using neural network. *«AVIA-2023» міжнародна науково-технічна конференція*: тези доп., 18-20 квіт, 2023. Київ, Україна. С. 10.1-10.5. URL: https://conference.nau.edu.ua/index.php/AVIA/AVIA2023/paper/view/9268/7607

10. Kucherov D., Dolgikh S., Myroshnychenko I., Poshyvailo O. Protocol and Logical Models for Robustness and Survivability in Autonomous UAV Systems. 2023 13th International Conference on Dependable Systems, Services and Technologies (DESSERT): conf. paper, 13-15 oct. 2023. Athens, Greece, 2023, P. 1-7, DOI: https://doi.org/10.1109/dessert61349.2023.10416491

11. Kucherov D., Poshyvailo O., Myroshnychenko I. Dynamics of UAV Discrete-Continuous Terminal Control System, *2023 IEEE 7th International Conference on Methods and Systems of Navigation and Motion Control (MSNMC)*: conf. paper, 24-27 oct, 2023 Kyiv, Ukraine. 2023. P. 66-69, DOI: https://doi.org/10.1109/msnmc61017.2023.10329065

12. Kucherov D., Shmelova T., Poshyvailo O., Tkachenko V., Myroshnychenko I., Ogirko I. Mathematical Model of Damping of UAV Oscillations in the Cargo Delivery Problem, *2023 IEEE 4th KhPI Week on Advanced Technology*

(KhPIWeek): conf. paper,, 02-06 Oct. 2023. Kharkiv, Ukraine, 2023, P. 1-6, DOI: https://doi.org/10.1109/khpiweek61412.2023.10312855

13. Myroshnychenko I., Kucherov D. Intellectual Control Systems for Aviation: Enhancing Autonomy and Safety. *Safetyin Aviation and Space Technologies. The Eleventh World Congress "AVIATION IN THE XXI-st CENTURY":* тези доп., 25-27 вер., 2024. Київ, Україна. С. 2.3.65-2.3.67 URL: https://jrnl.nau.edu.ua/index.php/congress/article/view/19225

14. Kucherov D., Myroshnychenko I., Khalimon N., Tkachenko V. Training of the Dynamic Systems Control: A Neural Network or a Learning Algorithm. *Artificial Intelligence and Applications*. 2025. DOI: https://doi.org/10.47852/bonviewAIA52025435