УДК 519.713:681.52

М. Б. ЄДИНОВИЧ, Т. О. КУЗЬМІНА, О. І. СОЛОГУБОВ
Херсонський національний технічний університет

МОДЕЛЮВАННЯ СИСТЕМІ УПРАВЛІННЯ БПЛА ІЗ ЗАСТОСУВАННЯМ МОДЕЛІ REINFORCEMENT LEARNING

У статті було досліджено найбільш поширені методи управління БПЛА із застосуванням як традиційних методів з використанням пропорційно-інтегрально-диференціального (ПІД) закону керування так і інтелектуальних систем. Застосування алгоритмів штучного інтелекту (ШІ), зокрема алгоритмів навчання з підкріпленням (Reinforcement Learning) забезпечує адаптивність до змінної динаміки та середовища. При створенні моделі БПЛА враховувалися основні показники системи керування БПЛА – амплітуда підйомної сили та різниця підйомної сили. Використання цих параметрів у моделюванні дозволяє забезпечити реалістичну поведінку дрона в двовимірному просторі, що дає змогу точно оцінити ефективність алгоритмів управління. Амплітуда підйомної сили відповідає за базову стабільність та виконання вертикальних завдань, тоді як різниця підйомної сили забезпечує можливість маневрування та досягнення заданих маршрутних точок. Була запропонована система оцінки продуктивності (scoring), яка базується на завданні навігації до випадкових точок у просторі, що представлені у вигляді «повітряних кульок». У ході досліджень було розроблено математичну модель та алгоритми управління, що враховують динаміку руху БПЛА у двовимірному просторі, з урахуванням впливу інерції, гравітації та тяги пропелерів. Було проведено моделювання з використанням алгоритмів DQN, SAC і SAC з модифікацією рівня диференціальної тяги. Виконано порівняльний аналіз ефективності зазначених підходів у різних сценаріях. Для тестування ефективності алгоритмів створено симуляційне середовище на основі мови програмування Python із використанням бібліотек NumPy, Matplotlib, Pygame та Stable-Baselines3. Середовище дозволяє моделювати завдання стабілізації польоту, навігації та уникнення перешкод.

З метою створення універсальної платформи для дослідження систем управління розроблено симуляційне середовище на основі мови Python. Це середовище дозволяє тестувати алгоритми в умовах, близьких до реальних, без необхідності використання дорогого обладнання.

Ключові слова: *БПЛА, дрон, ШІ, навчання з підкріпленням, контролер, агент, винагорода, імітація, тяга, спостереження.*

M. B. YEDYNOVYCH, T. O. KUZMINA, O. I. SOLOGUBOV
Kherson National Technical University

MODELING OF A UAV CONTROL SYSTEM USING THE REINFORCEMENT LEARNING MODEL

The article investigated the most common methods of UAV control using both traditional methods using the proportional-integral-differential (PID) control law and intelligent systems. The use of artificial intelligence (AI) algorithms, in particular reinforcement learning algorithms (Reinforcement Learning), ensures adaptability to changing dynamics and the environment. When creating a UAV model, the main indicators of the UAV control system were taken into account – Lift Amplitude and Lift Difference. The use of these parameters in modeling allows for realistic behavior of the drone in two-dimensional space, which makes it possible to accurately assess the effectiveness of control algorithms. The Lift Amplitude is responsible for basic stability and the performance of vertical tasks, while the Lift Difference provides the ability to maneuver and reach specified waypoints. A performance evaluation system (scoring) was proposed, which is based on the task of navigating to random points in space, represented in the form of «balloons». During the research, a mathematical model and control algorithms were developed that take into account the dynamics of UAV movement in two-dimensional space, taking into account the influence of inertia, gravity and propeller thrust. Simulation was carried out using the DQN, SAC and SAC algorithms with a modification of the differential thrust level. A comparative analysis of the effectiveness of the above approaches in different scenarios was performed. To test the effectiveness of the algorithms, a simulation environment based on the Python programming language was created using the NumPy, Matplotlib, Pygame and Stable-Baselines3 libraries. The environment allows you to model the tasks of flight stabilization, navigation and obstacle avoidance.

In order to create a universal platform for researching control systems, a simulation environment based on the Python language was developed. This environment allows you to test algorithms in conditions close to real ones, without the need to use expensive equipment.

Keywords: *UAV, drone, AI, reinforcement learning, controller, agent, reward, imitation, thrust, observation.*

https://doi.org/10.32782/mathematical-modelling/2026-9-1-10 ISSN 2618-0332 (print), 2618-0340 (online)

Постановка проблеми

Актуальність теми дослідження з огляду на виклики, що виникли в ході боротьби України з російською агресією надзвичайно велика. Застосування ворогом засобів РЕБ значно ускладнює керування БПЛА з управлінням по радіоканалу. Оптоволоконний зв'язок попри свою простоту має обмежену дальність. Використання засобів штучного інтелекту в системі наведення БПЛА значно підвищує бойову ефективність і захищеність літального апарату від засобів РЕБ противника [1].

Очевидно, що визначення траєкторії руху БПЛА у залежності від наявних перешкод, обмежень, пов'язаних з часом виконання завдання та запасу пального або заряду батареї суттєво впливає на ефективність використання БПЛА. Для спрощення задачі досліджувався рух літального апарату у двовимірному просторі. Такий підхід обумовлений достатністю двох координатних осей (X та Y) для вирішення поставлених завдань. У межах цього дослідження рух у 2D-просторі дозволяє оцінити ефективність алгоритмів стабілізації та навігації без потреби врахування додаткових факторів, характерних для тривимірного моделювання, таких як висота чи орієнтація в просторі. Застосування 2D-моделі також суттєво спрощує реалізацію симуляційного середовища і знижує обчислювальну складність завдання, що дозволяє проводити експерименти швидше та на менш потужних апаратних платформах.

Аналіз останніх досліджень і публікацій

У роботах [2–3] розглянуті системи автопілотування для БПЛА, що зазвичай складаються з «внутрішнього контуру», відповідального за стабілізацію та керування апаратом, та «зовнішнього контуру» для забезпечення цілей рівня місії (наприклад, навігація за точками маршруту). Системи керування польотом для БПЛА переважно реалізуються з використанням пропорційно-інтегрально-диференціальних (ПІД) систем керування. ПІД продемонстрували винятковий продуктивність за багатьох обставин, зокрема в контексті перегонів дронів, де точність та маневреність є ключовими. У стабільних умовах ПІД-контролер демонструє близьку до ідеальної продуктивність. Однак для роботи в непередбачуваних та суворих умовах потрібне більш складне керування. Інтелектуальні системи керування польотом є активною галуззю досліджень, що спрямовані на вирішення обмежень ПІД-керування останнім часом за допомогою навчання з підкріпленням (Reinforcement Learning – RL), яке мало успіх в інших галузях, зокрема робототехніці. Однак, під впливом випадкових факторів (наприклад, вітру, змінного корисного навантаження, провалу напруги), ПІД-контролер може бути далеко не оптимальним [4]. Інтелектуальна система управління забезпечує адаптивність до змінної динаміки та середовища. Розробка інтелектуальних систем керування польотом стає актуальним напрямом сучасних досліджень [5], зокрема, завдяки використанню штучних нейронних мереж, які є універсальними апроксиматорами та стійкі до шуму [2].

Методи онлайн-навчання [6] мають перевагу у вивченні динаміки БПЛА в режимі реального часу. Основним обмеженням онлайн-навчання є те, що система керування польотом знає лише про свій минулий досвід. Звідси випливає, що її продуктивність обмежена при впливі нової події. Навчання моделей офлайн з використанням навчання з учителем є проблематичним, оскільки потрібні дані походять з неточних представлень базової динаміки літака, наприклад, дані польоту з аналогічного апарата з використанням PID-керування, що може призвести до неоптимального алгоритму керування [7–9].

Альтернатива навчанню з учителем для створення офлайн-моделей відома як навчання з підкріпленням (RL). В RL агент отримує винагороду за кожну дію, яку він виконує в середовищі, з метою максимізувати винагороду з часом. Використовуючи RL, можна розробити оптимальну політику керування для БПЛА без будь-яких припущень щодо динаміки літака. У роботі [10] показано, що RL є ефективним для автопілотів БПЛА, забезпечуючи адекватне відстеження траєкторії.

https://doi.org/10.32782/mathematical-modelling/2026-9-1-10 ISSN 2618-0332 (print), 2618-0340 (online)

Глибоке навчання з підкріпленням (Deep reinforcement learning – DRL) демонструє гарні результати в робототехніці та плануванні місій на високому рівні, його застосування до планування місій на низькому рівні залишається недостатньо вивченим. Існує критична потреба в більш складному управлінні БПЛА для роботи в складних та непередбачуваних середовищах. У роботі [11] оцінюються алгоритми глибокого навчання з підкріпленням, таких як проксимальна оптимізація політики (Proximal Policy Optimization – PPO), глибокий детермінований градієнт політики (Deep Deterministic Policy Gradient – DDPG) та оптимізація політики довірчої області (Trust Region Policy Optimization – TRPO) для керування орієнтацією БПЛА. Встановлено, що контролер з комбінацією адаптивної оптимізації та DRL значно покращує характеристики польоту, забезпечуючи в середньому покращення часу підйому, коефіцієнта помилок на 19,9 %, 16,5 % відповідно, а також повністю стабільний політ.

Мета дослідження

Метою дослідження є аналіз поведінки системи управління БПЛА з використанням навчання з підкріпленням (Reinforcement Learning).

Виклад основного матеріалу дослідження

Для моделювання поведінки безпілотного літального апарата у двовимірному просторі ключовими аспектами є амплітуда підйомної сили та різниця підйомної сили. Ці параметри визначають здатність дрона змінювати висоту, стабілізувати своє положення та здійснювати маневри [11].

Амплітуда підйомної сили характеризує силу тяги, яка створюється пропелерами, і прямо впливає на швидкість вертикального переміщення апарата. У симуляції цей параметр моделюється як підсумкове значення тяги лівого і правого пропелерів, що дозволяє точно відтворити процес підйому чи опускання дрона.

Різниця підйомної сили між роторами, у свою чергу, визначає нахил апарата, що впливає на горизонтальне переміщення. Цей показник є критично важливим для маневреності дрона, оскільки дозволяє змінювати його траєкторію та орієнтацію в просторі. У 2D-середовищі різниця підйомної сили відображає дисбаланс тяги між лівим і правим пропелерами, завдяки чому дрон може нахилятися та рухатися в потрібному напрямку.

Використання цих параметрів у моделюванні дозволяє забезпечити реалістичну поведінку дрона в двовимірному просторі, що дає змогу точно оцінити ефективність алгоритмів управління. Амплітуда підйомної сили відповідає за базову стабільність та виконання вертикальних завдань, тоді як різниця підйомної сили забезпечує можливість маневрування та досягнення заданих маршрутних точок.

Для оцінки різних підходів до керування ми використовуємо 2D середовище, що моделює фізику твердого тіла, в якому 2D квадрокоптер повинен орієнтуватися на випадково згенеровані точки, представлені у вигляді повітряних кульок досягаючи їх за певну кількість часу, що допоможе нам вирахувати ефективність і продуктивність управління дроном різними агентами.

У стимуляційному середовищі динаміка польоту дрона моделюється з частотою 60 кроків на секунду. Це означає, що за кожну секунду симуляція обчислює новий стан дрона, включаючи його прискорення, швидкість і позицію 60 разів. Така частота дозволяє забезпечити високу точність і стабільність симуляції, що є важливим для коректного моделювання фізики польоту.

Важливим елементом симуляції є система оцінки продуктивності (scoring), яка базується на завданні навігації до випадкових точок у просторі, що представлені у вигляді «повітряних кульок». Алгоритм оцінювання полягає у наступному:

– дрон отримує 1 бал щоразу, коли досягає цільової точки;

– після досягнення цілі нова точка з’являється у випадковому місці в межах симуляційного середовища;

https://doi.org/10.32782/mathematical-modelling/2026-9-1-10 ISSN 2618-0332 (print), 2618-0340 (online)

– якщо дрон відхиляється занадто далеко від цільової точки, це вважається крахом, і апарат «відроджується» на початковій позиції з паузою у 3 секунди.

Завдання дрона – зібрати якомога більше балів протягом обмеженого часу, який у симуляції складає 100 секунд. Загальний результат визначається кількістю досягнутих цільових точок (повітряних кульок) за цей час. Така система оцінки дозволяє порівнювати ефективність різних підходів до управління, зокрема ручного керування оператором, PID-контролера та алгоритмів навчання з підкріпленням.

Частота 60 кроків за секунду забезпечує плавну динаміку польоту та дозволяє точно фіксувати зміни у положенні та швидкості дрона, що є особливо важливими при аналізі його реакції на вхідні команди або зміну умов управління.

У першій спробі навчання агента навчання з підкріпленням використовувався алгоритм DQN (Deep Q-Network) із бібліотеки Stable-Baselines3 [12]. Метою було перевірити працездатність процесу навчання та встановити базовий рівень ефективності для подальших експериментів.

1. Утримання стабільної тяги: $T_1 = 0.04$, $T_r = 0.04$.
2. Підйом: $T_1 = 0.08$, $T_r = 0.08$.
3. Зменшення тяги: $T_1 = 0$, $T_r = 0$.
4. Нахил ліворуч: $T_1 = 0.0394$, $T_r = 0.0406$.
5. Нахил праворуч: $T_1 = 0.0394$, $T_r = -0.0406$.

Тип дій алгоритму DQN: дискретний простір дій. Дії були чітко визначені як фіксовані значення, що відповідають конкретним змінам тяги.

Агент обирає дію кожні 5 кроків симуляції, і обрана дія виконується протягом цих 5 кроків. Агент вивчає оптимальні стратегії шляхом оцінки функції $Q(s, a)$, яка представляє очікувану винагороду за виконання дії a у стані s. DQN оновлює нейронну мережу використовуючи рівняння Беллмана для апроксимації Q-значень (очікуваних винагород) для кожної дії:

$$Q(s,a) = r + \gamma \max_{a'} Q(s',a'),$$

де r – винагорода за виконану дію,

γ – коефіцієнт дисконтування, що визначає важливість майбутніх винагород,

α – швидкість навчання.

Досвід (стан, дія, винагорода, новий стан) зберігається у буфері, щоб навчатися на рандомізованих вибірках, наближаючи кореляцію між даними.

Навчання проводилося на 1 мільйоні кроків симуляції. Стан агента (спостереження) та результати його дій відстежувалися за допомогою інструменту Weights and Biases [13].

Навчання проходило у кілька етапів.

1. Етап початкового навчання (0–400 тис. кроків):

На початку навчання рівень дослідження середовища був дуже високим, що змушувало агента часто виконувати випадкові дії. Це призводило до частих аварій та негативної винагороди, яка досягала –1000 через аварії.

2. Середній етап (400–800 тис. кроків):

Зі зменшенням коефіцієнта дослідження агент поступово почав навчатися уникати аварій та показував середню винагороду близько –400.

Однак нестабільність у навчанні призводила до періодичних провалів між 600 тис. та 800 тис. кроками.

3. Завершальний етап (приблизно 1 млн. кроків):

Агент досяг максимальної середньої винагороди 345 завдяки вдосконаленій стратегії, але цей прогрес був нетривалим – після цього агент знову почав часто зазнавати невдач (рис. 1).

Проведені дослідження підтвердили доцільність використання швидкості зміни вихідного сигналу досліджуваного об’єкту для визначення сталих часу і запізнення. Запропонований

https://doi.org/10.32782/mathematical-modelling/2026-9-1-10 ISSN 2618-0332 (print), 2618-0340 (online)

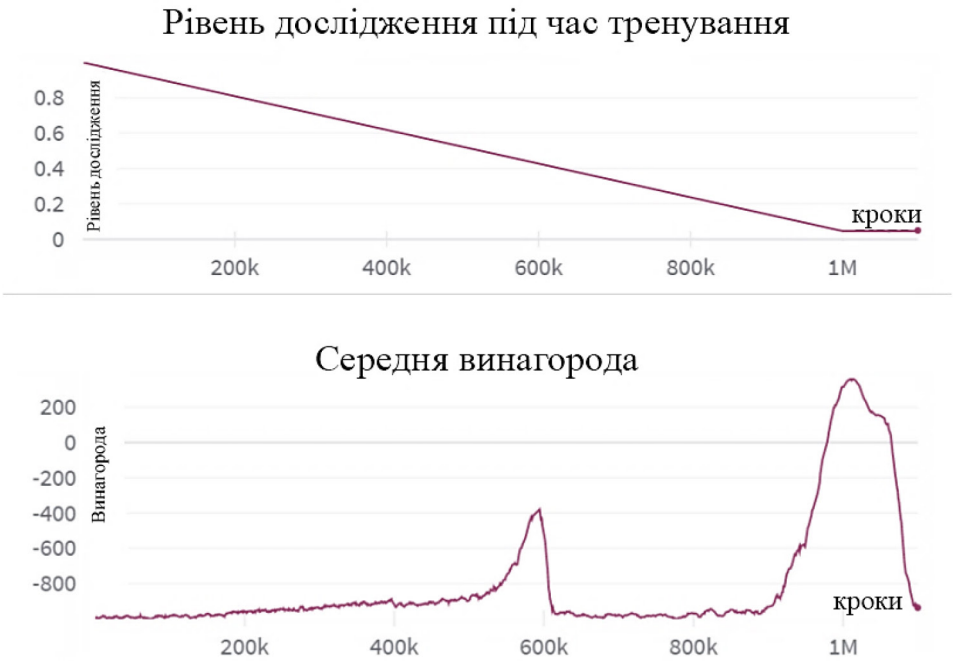

Рис. 1. Результати відстеження із застосуванням алгоритму DQN

спосіб дозволяє значно підвищити ефективність відомих графоаналітичних методів ідентифікації. У ході досліджень з'ясувалося, що шуми, присутні у сигналах датчиків можуть суттєво вплинути на точність визначення параметрів об'єкта управління. Для зменшення впливу цих шумів необхідно застосовувати фільтри нижніх частот.

Попри обмеження, використання DQN дозволило агенту продемонструвати базову здатність до навчання в нестабільному середовищі. Досягнуті результати підтверджують ефективність підходу навчання з підкріпленням, але вказують на необхідність переходу до більш стабільних алгоритмів (наприклад, SAC) або розширення простору дій для покращення управління дроном.

SAC є набагато досконалішим алгоритмом і на сьогоднішній день є найсучаснішим, коли мова йде про безперервні дії. Обраний простір дій наступний: вихід агента – це два плаваючих значення $дія_0$ і $дія_1$ (у коді програми представлено як $action_0$ і $action_1$). $Дія_0$ представляє амплітуду тяги, прикладену до обох роторів. $Дія_1$ представляє відносну різницю тяги, прикладену до обох роторів. Сили тяги можуть бути отримані з дій за допомогою наступних формул (перетворення дій у рівняння сил):

$$T_1 = дія_0 \cdot 0{,}04 + дія_1 \cdot 0{,}0006 + 0{,}04$$

$$T_1 = дія_0 \cdot 0{,}04 - дія_1 \cdot 0{,}0006 + 0{,}04$$

Для програмної реалізації використовувалась бібліотека Stable-Baselines3, клас SAC.

Тип дій: безперервний простір дій. SAC генерує два плаваючих значення ($дія_0$ і $дія_1$). Результати моделювання наведені на рис. 2.

Навчання відбувалося з тривалістю в 3,3 мільйони кроків. Вже після 50 тисяч кроків агент показав винагороду вище 0, що означає, що він навчився уникати аварій. Це значно перевершує результат DQN. Винагороди продовжували зростати впродовж тренування, демонструючи, що SAC успішно навчається та адаптується до середовища.

Середня винагорода стабілізувалася на рівні 950 балів на епізод, що еквівалентно збору приблизно 9 цілей (шарів) кожні 20 секунд.

SAC продемонстрував набагато кращу продуктивність порівняно з DQN, як показано на графіку.

https://doi.org/10.32782/mathematical-modelling/2026-9-1-10 ISSN 2618-0332 (print), 2618-0340 (online)

Рис. 2. Порівняння відстеження винагороди SAC алгоритму з DQN

Під час аналізу було виявлено, що $дія_1$ часто досягала своїх граничних значень (–1 або 1). Агент працював на межі своїх можливостей, що могло обмежити його продуктивність. Для перевірки було заплановано новий запуск тестування з удосконаленою конфігурацією.

У третьому запуску було використано алгоритм SAC, але з модифікацією рівня диференціальної тяги для підвищення агресивності обертання дрона (рис. 3). Це дозволило агенту швидше та ефективніше змінювати напрямок польоту.

Для підвищення ефективності обертання дрона було змінено коефіцієнт диференціальної тяги у формулах, що описують тягу лівого та правого пропелерів:

$$T_l = дія_0 \cdot 0{,}04 + дія_1 \cdot 0{,}003 + 0{,}04$$
$$T_r = дія_0 \cdot 0{,}04 - дія_1 \cdot 0{,}003 + 0{,}04$$

Збільшення коефіцієнта 0,003 у диференціальній тязі дозволяє дрону обертатися агресивніше та швидше реагувати на необхідність зміни напряму. Навчання відбувалося з тривалістю в 5 мільйони кроків.

Рис. 3. Порівняння відстеження винагороди SAC алгоритму з удосконаленим алгоритмом SAC_2

https://doi.org/10.32782/mathematical-modelling/2026-9-1-10 ISSN 2618-0332 (print), 2618-0340 (online)

Агент SAC навчився використовувати нову можливість швидкого обертання для точнішого досягнення цільових точок (кульок).

За результатами навчання на графіку середня винагорода стабілізувалася приблизно на рівні 1200 балів за епізод. Це відповідає збору приблизно 12 кульок кожні 20 секунд, що є значним покращенням продуктивності.

Як видно з рис. 3, SAC_2 демонструє стабільний прогрес із мінімальними коливаннями винагороди на пізніх етапах навчання.

Висновки

Результати проведених досліджень показали, що модифікація формул для вищої диференціальної тяги дозволяє агенту SAC досягати кращої продуктивності та підвищити ефективність керування дроном. Агент навчається швидше реагувати на необхідність зміни напряму та демонструвати кращі результати порівняно з іншими агентами, зокрема PID-контролером та людиною.

Це підтверджує, що адаптація параметрів керування у поєднанні з SAC дозволяє досягти високої продуктивності у створеному 2D середовищі.

Список використаної літератури

1. Литвиненко М. І., Ленець, В. Г., Гармаш Н. В. Шульга В. В. Аспекти впровадження штучного інтелекту у військовій справі. *Збірник наукових праць Харківського національного університету Повітряних Сил*. 2024. С. 13–18. DOI: https://doi.org/10.30748/zhups.2024.80.02
2. Koch W., Mancuso R., West R., Bestavros A. 2019. Reinforcement Learning for UAV Attitude Control. *ACM Transactions on Cyber-Physical Systems*. 2019. Vol. 3. №. 2. Article 22, 21 pages. https://doi.org/10.1145/3301273
3. ArduPilot Copter URL: https://ardupilot.org/copter/index.html (Дата звернення 18.11.25).
4. Maleki K. N., Ashenayi K., Hook L. R., Fuller J. G., Hutchins N. A reliable system design for nondeterministic adaptive controllers in small UAV autopilots. *2016 IEEE/AIAA 35th Digital Avionics Systems Conference* (*DASC*'16). Sacramento, 25-29 September 2016. CA, USA, 2016, pp. 1–5, DOI: https://doi.org/10.1109/DASC.2016.7778103
5. Santoso F., Garratt M. A., Anavatti S. G. State-of-the-art intelligent flight control systems in unmanned aerial vehicles. *IEEE Transactions on Automation Science and Engineering*. 2017. Vol. 15, № 2, P. 613–627. DOI: https://doi.org/10.1109/TASE.2017.2651109
6. Dierks T., Jagannathan S. 2010. Output feedback control of a quadrotor UAV using neural networks. *IEEE Transactions on Neural Networks*. 2010. Vol. 21, № 1. P. 50–66. DOI: https://doi.org/10.1109/TNN.2009.2034145
7. Bobtsov A., Guirik A., Budko M., Budko M. Hybrid parallel neuro-controller for multiro-tor unmanned aerial vehicle. *2016 8th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops* (*ICUMT'16*), 18–20 October 2016. Lisbon, Portugal, 2016. P. 1–4. DOI: https://doi.org/10.1109/ICUMT.2016.7765223
8. Shepherd III J. F. Tumer K. Robust neuro-control for a micro quadrotor. In *Proceedings of the 12thAnnual Conference on Genetic and Evolutionary Computation*. (*GECCO'10*). ACM, 2010. New York, NY, 1131–1138. https://doi.org/10.1145/1830483.1830693
9. Miglino O., Lund H. H., Nolfi S. Evolving mobile robots in simulated and real environ-ments. *Artificial Life* 1995. Vol. 2. № 4, 417–434. DOI: https://doi.org/10.1162/artl.1995.2.4.417
10. Hwangbo J., Sa I., Siegwart R., Hutter M. Control of a quadrotor with reinforcementlearning. *IEEE Robotics and Automation Letters*. 2017. Vol. 2. № 4. P. 2096–2103. https://doi.org/10.1109/LRA.2017.2720851
11. Zorain M., Khan F. S., Hasanv N., Mohy Ud Din Z., Zeb Gul J. Deep reinforcement learning for UAV attitude control via adaptive gain optimization *Applied Intelligence*. 2025. Vol. 55. Issue 17. P. 1092. https://doi.org/10.1007/s10489-025-06978-1

12. Stable-Baselines3 Docs – Reliable Reinforcement Learning Implementations URL: https://stable-baselines3.readthedocs.io/en/v1.0/ (Дата звернення 20.11.25).
13. Weights & Biases AI developer platform. URL: https://wandb.ai/site/ (Дата звернення 20.11.25).

References

1. Lytvynenko M. I., Lenets, V. G., Garmash N. V., & Shulga V. V. Aspects of the Implementation of Artificial Intelligence in Military Affairs. [Aspects of the Implementation of Artificial Intelligence in Military Affairs]. *Collection of scientific papers of the Kharkiv National University of the Air Force*. 2024. Pp. 13–18. DOI: https://doi.org/10.30748/zhups.2024.80.02. [in Ukrainian].
2. Koch W., Mancuso R., West R., & Bestavros A. (2019). Reinforcement Learning for UAV Attitude Control. *ACM Transactions on Cyber-Physical Systems*, 3(2). Article 22, 21 pages. https://doi.org/10.1145/3301273 [in English].
3. ArduPilot Copter URL: https://ardupilot.org/copter/index.html (Date of access 18.11.25) [in English].
4. Niki Maleki K., Ashenayi K., Hook L. R., Fuller J. G., & Hutchins N. (2016). A reliable system design for nondeterministic adaptive controllers in small UAV autopilots. In *Proceedings of the 2016 IEEE/AIAA 35th Digital Avionics Systems Conference* (*DASC'16*). Sacramento, CA, USA. DOI: https://doi.org/10.1109/DASC.2016.7778103. [in English].
5. Santoso F., Garratt M. A., & Anavatti S. G. (2017). State-of-the-art intelligent flight control systems in unmanned aerial vehicles. *IEEE Transactions on Automation Science and Engineering*, 15(2), 613–627. DOI: https://doi.org/10.1109/TASE.2017.2651109 [in English].
6. Dierks T., & Jagannathan S. (2010). Output feedback control of a quadrotor UAV using neural net-works. *IEEE Transactions on Neural Networks* 21 (1), 50–66. DOI: https://doi.org/10.1109/TNN.2009.2034145. [in English].
7. Bobtsov A., Guirik A., Budko M., & Budko M. (2016). Hybrid parallel neuro-controller for multiro-tor unmanned aerial vehicle. In *Proceedings of the 2016 8th International Congress on Ultra Modern Telecommunicationsand Control Systems and Workshops* (*ICUMT'16*). IEEE, Lisbon, Portugal.,https://doi.org/10.1109/ICUMT.2016.7765223 [in English].
8. Shepherd III J. F. & Tumer K. (2010). Robust neuro-control for a micro quadrotor. In *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation*. (*GECCO'10*). ACM, New York, NY, 1131–1138. https://doi.org/10.1145/1830483.1830693. [in English].
9. Miglino O., Lund H. H., & Nolfi S. (1995). Evolving mobile robots in simulated and real environ-ments. *Artificial Life*, 2 (4), 417–434. DOI: https://doi.org/10.1162/artl.1995.2.4.417. [in English].
10. Hwangbo J., Sa I., Siegwart R., & Hutter M. (2017). Control of a quadrotor with reinforcementlearning. *IEEE Robotics and Automation Letters* 2, 4 (2017), 2096–2103. https://doi.org/10.1109/LRA.2017.2720851. [in English].
11. Zorain M., Khan F.S., Hasanv N., Mohy Ud Din Z., & Zeb Gul J. (2025). Deep reinforcement learning for UAV attitude control via adaptive gain optimization *Applied Intelligence*. 55 (17), 1092. https://doi.org/10.1007/s10489-025-06978-1. [in English].
12. Stable-Baselines3 Docs – Reliable Reinforcement Learning Implementations URL: https://stable-baselines3.readthedocs.io/en/v1.0/ (Date of access 20.11.25). [in English].
13. Weights & Biases AI developer platform URL: https://wandb.ai/site/ (Date of access 20.11.25). [in English].

Єдинович Михайло Борисович – к.т.н., доцент, доцент кафедри автоматики, робототехніки і мехатроніки Херсонського національного технічного університету. E-mail: myedyno@gmail.com, ORCID: 0000-0002-6113-1923.

Кузьміна Тетяна Олегівна – д.т.н., професор, професор кафедри товарознавства, стандартизації та сертифікації Херсонського національного технічного університету. E-mail: edenkuz@gmail.com, ORCID: 0000-0002-6113-1923.

Сологубов Олексій Ігорович – магістр кафедри автоматизації, робототехніки і меха-троніки Херсонського національного технічного університету. E-mail: revelated17@gmail.com, ORCID: 0009-0004-4203-3350.

Yedynovych Mykhailo Borysovych – Candidate of Technical Sciences, Associate Professor, Associate Professor at the Department of Automation, Robotics and Mechatronics of the Kherson National Technical University. E-mail: myedyno@gmail.com, ORCID: 0000-0002-6113-1923.

Kuzmina Tetyana Olehivna – Doctor of Technical Sciences, Professor, Professor at the Department of Commodity Science, Standardization and Certification of the Kherson National Technical University. E-mail: edenkuz@gmail.com, ORCID: 0000-0002-6113-1923.

Sologubov Oleksii Ihorovych – Master's Student at the Department of Automation, Robotics and Mechatronics of the Kherson National Technical University. E-mail: revelated17@gmail.com, ORCID: 0009-0004-4203-3350.

Дата першого надходження статті до видання: 05.03.2026
Дата прийняття статті до друку після рецензування: 07.04.2026
Дата публікації (оприлюднення) статті: 01.07.2026