УДК 004.8'231:623.746.9
DOI https://doi.org/10.32782/tnv-tech.2025.2.12

ПОРІВНЯННЯ АДАПТИВНОСТІ RL-АЛГОРИТМІВ ДЛЯ УНИКНЕННЯ ЗІТКНЕНЬ БПЛА

Копилець М. М. *– аспірант кафедри інформаційних систем та мереж Національного університету «Львівська політехніка»*
ORCID ID: 0009-0004-5823-9871

У цій роботі досліджується адаптивність алгоритмів навчання з підкріпленням (RL), зокрема DQN, PPO, SAC, TD3, DDPG та A2C, у змодельованому тривимірному середовищі, що імітує сценарії уникнення зіткнень безпілотних літальних апаратів (БПЛА). Метою дослідження є порівняння здатності кожного алгоритму узагальнювати отримані знання та використовувати їх у нових непередбачуваних умовах польоту.

Експериментальна частина складається з двох етапів. На першому етапі дрони тренуються в умовах, коли їхні траєкторії спрямовуються назустріч один одному з фіксованими початковими параметрами швидкості та положення. На другому етапі кожний алгоритм проходить валідацію в сценаріях, що виходять за межі тренування, з випадковими змінами напрямків польоту та швидкісних характеристик. Такий підхід дозволяє оцінити стабільність поведінки моделей у нових, непередбачуваних умовах. У результатах дослідження порівнюються показники успішного уникнення зіткнень та якість маневрування для заданих алгоритмів. Отримані дані демонструють, що деякі алгоритми, такі як SAC та TD3, показують вищий рівень стабільності в умовах сильних коливань траєкторій, тоді як DQN і PPO можуть бути менш стійкими на непередбачуваних етапах випробувань. A2C та DDPG виявилися посередніми за показниками успішності визначеними в ході експерименту.

Запропоноване дослідження надає цінну інформацію для практичної реалізації систем автономного управління БПЛА в динамічних та невизначених середовищах. Отримані висновки можуть бути використані при розробці адаптивних контролерів, що здатні швидко коригувати поведінку дрона у випадку виникнення непередбачуваних ситуацій. У подальших дослідженнях слід підключити сенсорні системи, максимально наближені до реальних умов польоту, та перевірити, як це вплине на адаптивність алгоритмів. Це забезпечить подальше підвищення надійності та безпеки безпілотних систем у реальних польотних місіях.

Ключові слова: *DQN, PPO, SAC, TD3, DDPG, A2C, навігація дронів, уникнення зіткнень, навчання з підкріпленням.*

Kopylets M. M. Comparison of RL algorithm adaptivity for UAV collision avoidance

This paper investigates the adaptivity of reinforcement learning (RL) algorithms, specifically DQN, PPO, SAC, TD3, DDPG, and A2C, in a simulated three-dimensional environment that mirrors unmanned-aerial-vehicle (UAV) collision-avoidance scenarios. The aim of the study is to compare each algorithm's ability to generalize acquired knowledge and apply it in new, unpredictable flight conditions.

The experimental section consists of two stages. In the first stage, drones are trained in conditions where their trajectories are directed toward one another, with fixed initial speed and position parameters. In the second stage, each algorithm is validated in scenarios that lie outside the training domain, featuring random changes in flight direction and velocity characteristics. This approach makes it possible to assess the stability of model behavior in newly encountered, unforeseen conditions. The study compares the rates of successful collision avoidance, reaction speed, and maneuver quality for each algorithm. The data show that some algorithms, such as SAC and TD3, exhibit higher stability under strong trajectory fluctuations, whereas DQN and PPO may be less robust during unpredictable test phases. A2C and DDPG proved intermediate in both success rate and decision-making time.

The proposed research provides valuable information for the practical implementation of autonomous UAV control systems in dynamic and uncertain environments. The findings can be used to develop adaptive controllers capable of rapidly correcting drone behavior when unforeseen situations arise. Future studies should incorporate sensor systems that closely

replicate real flight conditions and assess how they affect algorithm adaptivity. This will further enhance the reliability and safety of unmanned systems in real flight missions.

Key words: *DQN, PPO, SAC, TD3, DDPG, A2C, drone navigation, collision avoidance, reinforcement learning.*

Постановка проблеми. Безпілотні літальні апарати (БПЛА) [1] набувають дедалі більшої популярності у комерційному, цивільному та військовому секторах. Їх застосовують для аерофотозйомки, спостереження, доставки вантажів та інших завдань. Чим складнішими є польотні місії тим надійнішою повинна бути навігація БПЛА й ефективне уникнення перешкод стає критично важливим для безпечної та ефективної роботи. Навчання з підкріпленням (RL) [2] пропонує перспективний підхід до вдосконалення навігації БПЛА, дозволяючи системам навчатися на власному досвіді й адаптуватися до нових умов. Проте жодна окрема модель RL не є універсальною. Алгоритми Deep Q-Network (DQN) [3], Proximal Policy Optimization (PPO) [4], Soft Actor-Critic (SAC) [5], Twin Delayed Deep Deterministic Policy Gradient (TD3) [6], Deep Deterministic Policy Gradient (DDPG) [7] та Advantage Actor-Critic (A2C) [8] мають як переваги, так і обмеження, показуючи добрі результати у контрольованих умовах, вони можуть втрачати ефективність перед несподіваними викликами.

У цьому дослідженні використано тривимірне симуляційне середовище, де дрони навчаються в сценаріях із зустрічними БПЛА, що створює реалістичну задачу уникнення зіткнень. У ході проведення експериментів оцінюється кожен алгоритм, порівнюючи частку успішних уникнень та характер поведінки у ситуаціях, що відрізняються від тренувальних умов. Аналіз отриманих результатів дає змогу оцінити адаптивність кожного RL-алгоритму й визначити методи, найстійкіші до динамічних умов.

Аналіз досліджень і публікацій. Дослідження проблеми уникнення зіткнень БПЛА [9] загалом зосереджуються на двох ключових підходах: класичних алгоритмах планування руху, що спираються на явні математичні моделі, та новітніх методах, які використовують навчання з підкріпленням. Ранні рішення, такі як пошук по сітці та потенційні поля, працюють ефективно у статичних або слабо динамічних середовищах. Проте ці традиційні техніки виявляються малопридатними в умовах які швидко змінюються, оскільки потребують детальних наперед відомостей про перешкоди й погано адаптуються до змін. Через це RL дедалі активніше застосовують у навігації БПЛА, агенти формують політики уникнення зіткнень, багаторазово взаємодіючи з модельованими чи реальними сценаріями. В роботі описуються алгоритми, що базуються на Q-learning (DQN, Double DQN, prioritized replay) для дискретного керування, а також методи на основі градієнта (DDPG, PPO, SAC, TD3) для безперервних дій. Перебудовуючи політики безпосередньо з даних від середовища, ці підходи демонструють значний потенціал у неструктурованих середовищах з поведінкою, яку класичним методам важко передбачити.

У роботі [10] автори оцінюють DQN, PPO та Soft Actor-Critic (SAC) для уникнення перешкод БПЛА у змодельованому тривимірному середовищі. Дослідження охоплює як стаціонарні, так і рухомі перешкоди, підкреслюючи важливість вибору відповідної RL-стратегії під конкретні виклики. SAC демонструє найкращі результати там, де потрібна гнучка реакція, тоді як DQN з використанням вибіркової пам'яті (replay buffer) показує кращу ефективність. Натомість робота відзначає, що алгоритм PPO має певні труднощі з обробкою складних динамічних 3D-сценаріїв. Ці висновки напряму стосуються розуміння того, як різні алгоритми

поводяться у близьких до реальності задачах уникнення зіткнень, подібних до тих, що розглядаються в поточній роботі.

Інша стаття [11] досліджує роботу DQN і PPO у двовимірному середовищі, аналізуючи вплив функцій активації Tanh, Sigmoid, ReLU та Leaky ReLU на швидкість навчання й навігаційну поведінку моделей. За показниками середньої кількості кроків за епізод, відсотка повних зупинок, кількості поворотів і середньої винагороди автори окреслюють компроміси у продуктивності. Наприклад, одна функція активації може забезпечувати плавніший і безпечніший рух на відкритих ділянках, у той час як інша підвищує маневровість у вузьких проходах. Отримані результати підкреслюють, що вибір функції активації істотно впливає на процес ухвалення рішень і загальну ефективність дрона.

Мета дослідження. Метою дослідження є налаштування реалістичного тривимірного симуляційного середовища для дослідження навігаційних викликів БПЛА, тренування низки моделей глибинного навчання з підкріпленням, які покращують здатність дронів уникати зіткнень, та експериментальна перевірка їхньої адаптивності шляхом зміни траєкторії наближення іншого дрона.

Виклад основного матеріалу. Середовище симуляції було розроблено за допомогою мови Python, яка відома своєю гнучкістю та широкою екосистемою бібліотек, що робить її ідеальною для швидкого прототипування та створення складних динамічних моделей. Для графіки та інтерактивності було застосовано бібліотеку PyBullet [12] – фізичний рушій, який забезпечує реалістичну динаміку та перевірку зіткнень. Штучний інтелект реалізовано через бібліотеку stable_baselines3 [13, 14], яка містить повний набір алгоритмів навчання з підкріпленням. Готові політики нейронних мереж спрощують тренування і забезпечують відтворюваність під час розробки складних моделей навігації БПЛА.

За допомогою PyBullet було створено симуляційне 3D-середовище з двома квадрокоптерами. Кожен з них описано у форматі URDF (Unified Robot Description Format) – XML-файлі, що задає геометрію, системи координат і текстури роботизованої моделі. На рисунку 1 показано загальний вигляд сцени, обидва дрони на початку епізоду спрямовані один до одного. В експерименті обидва апарати летять назустріч, один рухається фіксованою прямою траєкторією без корекцій, другий контролюється навчальною моделлю, яка адаптує свій курс, щоб уникнути зіткнення.

Керування задається швидкісними координатами у вигляді трійки (v_x, v_y, v_z). Компонента v_x залишається сталою, що забезпечує рух обох дронів уперед із визначеною швидкістю, тоді як v_y і v_z змінюються, дозволяючи відхилятися ліворуч/праворуч і коригувати висоту відповідно. Керований дрон постійно отримує відносне положення опонента, завдяки чому обчислює оптимальні v_y та v_z для запобігання негайній колізії. Координати дронів оновлюються щокроку, тож курс можна коригувати в реальному часі.

Епізод стартує з розміщення дронів на протилежних кінцях сцени та завершується, коли:

- керований дрон стикається з іншим дроном або з підлогою;
- керований дрон безпечно пролітає простір і досягає стартової позиції опонента, що означає успішне уникнення зіткнення.

В таких умовах модель поступово навчається використовувати v_y і v_z для точних бокових і вертикальних маневрів, гарантуючи безпечний проліт повз зустрічний дрон без наперед заданих маршрутів чи карт перешкод.

У ході експерименту агент працює в одному з двох просторів дій, залежно від використаного алгоритму. Якщо застосовується DQN, простір дій дискретний

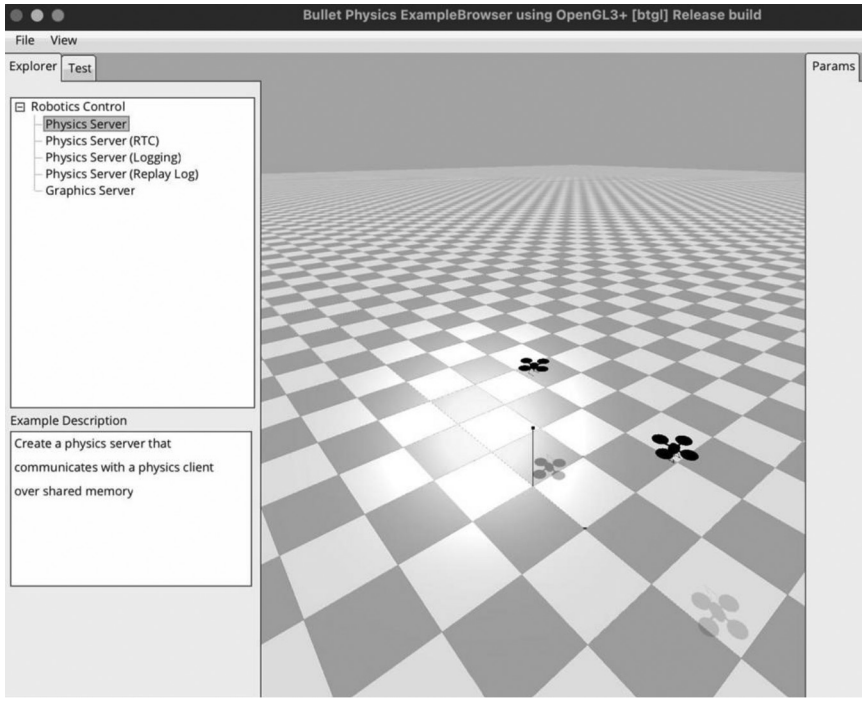

Рис. 1. Симуляційне середовище

і агент на кожному кроці обирає одну з фіксованого набору категоріальних дій. Для алгоритмів PPO, SAC, TD3, DDPG чи A2C простір дій безперервний, що дає змогу виконувати плавні мікрокорекції. Дискретна дія задається цілим числом $a \in \{0, 1, \ldots, 8\}$. Це число перетворюється на зміни швидкості в напрямках y та z. Математично це можна описати так:

$$\begin{aligned}
a = 0 &\rightarrow \Delta y = 0, \quad \Delta z = 0,\\
a = 1 &\rightarrow \Delta y = +1, \quad \Delta z = 0,\\
a = 2 &\rightarrow \Delta y = -1, \quad \Delta z = 0,\\
a = 3 &\rightarrow \Delta y = 0, \quad \Delta z = +1,\\
a = 4 &\rightarrow \Delta y = 0, \quad \Delta z = -1,\\
a = 5 &\rightarrow \Delta y = +1, \quad \Delta z = +1,\\
a = 6 &\rightarrow \Delta y = +1, \quad \Delta z = -1,\\
a = 7 &\rightarrow \Delta y = -1, \quad \Delta z = +1,\\
a = 8 &\rightarrow \Delta y = -1, \quad \Delta z = -1.
\end{aligned} \tag{1}$$

На відміну від дискретного простору дій, у безперервній версії середовища дія задається двовимірним вектором $a = (a_y, a_z)$, кожна компонента якого перебуває в діапазоні [–1; 1]. Щоб перетворити ці значення на фактичні зміни швидкості,

середовище просто додає a_y і a_z до поточних швидкісних компонент дрона вздовж осей y та z. Узагальнено це можна записати так:

$$\text{Дискретний простір дій: } a \in \{0, 1, \ldots, 8\}. \quad (2)$$

$$\text{Безперервний простір дій: } a = (a_y, a_z),\ a_y, a_z \in [1; 1]. \quad (3)$$

У всіх випадках швидкість уздовж осі x лишається сталою, з протилежними знаками для обох дронів, завдяки чому вони летять назустріч, а агент керує лише боковими та вертикальними зміщеннями.

Простір спостережень містить шість змінних, що описують швидкість керованого дрона та позицію іншого дрона відносно нього. Перші три компоненти кодують швидкість керованого дрона, нормалізовану так, щоб діапазон відповідав [–1; 1]. Решта три компоненти визначають відносну позицію другого дрона щодо першого, також нормалізовану до [–1; 1]. Формально, якщо $v_1 = (v_{1x}, v_{2x}, v_{3x})$ – швидкість керованого дрона, а $p_1, p_2 \in \mathbb{R}^3$ – їхні відповідні позиції, тоді вектор спостережень $o \in \mathbb{R}^6$ записується так:

$$o = (\Phi(v_1), \Psi(p_2 - p_1)), \quad (4)$$

де оператор Φ нормалізує кожну компоненту швидкості, щоб умістити її в [–1; 1], а Ψ нормалізує відносні координати із тією самою метою. Таке перетворення утримує вхідні дані в стабільному діапазоні, полегшуючи збіжність алгоритмів навчання.

Визначальним елементом тренування є функція винагороди, що спрямовує дії агента та формує політику. На кожному кроці дрон отримує винагороду залежно від того, наскільки добре він дотримується бажаної траєкторії і чи відбулося зіткнення або успішне розходження. Агент зазнає невеликого штрафу, пропорційного відхиленню від початкової траєкторії, яка передбачає зміну лише координати x стартової позиції (x_0, y_0, z_0). Тому якщо (y_1, z_1) – поточна позиція керованого дрона, то штраф задається як:

$$\text{штраф} = |y_0 - y_1| + |z_0 - z_1|. \quad (5)$$

Цей штраф віднімається від винагороди агента на кожному кроці симуляції:

$$r_t = -\text{штраф}. \quad (6)$$

Крім того, два ключові випадки різко змінюють значення винагороди. Зіткнення з іншим дроном або з поверхнею негайно нараховує агенту −1000 балів і завершує епізод:

$$r_t = -1000, \text{ завершення епізоду}. \quad (7)$$

Натомість, якщо керований дрон успішно пролітає простір і перетинає стартову позицію другого дрона вздовж осі x без зіткнення, він отримує +1000 балів, після чого епізод також завершується:

$$r_t = +1000, \text{ завершення епізоду}. \quad (8)$$

Малий штраф на кожному кроці й великі ±1000 балів у фіналі спонукають дрон рухатися безпечно. Агент успішно навчається обходити зустрічний дрон і уникати зіткнень.

Після завершення навчання модель перевіряється у трьох різних сценаріях, щоб оцінити її здатність узагальнювати знання поза умов тренування. Для кожного сценарію запускається 800 симуляцій із фіксацією кількості успішних і невдалих

маневрів уникнення. Порівняння цих показників демонструє, наскільки модель пристосовується до змін у середовищі.

Перший сценарій майже повторює тренувальний, зустрічний дрон летить прямолінійно, але його стартові координати випадково зміщуються по осях *y* та *z*. Це примушує навчений дрон реагувати на невеликі зміни кута зближення й показує, чи здатна політика коригуватися при незначних відхиленнях траєкторії. Другий сценарій зберігає ту саму початкову варіативність, але замість прямолінійного польоту вводить зигзагоподібну траєкторію зустрічного дрона. Часті бокові зміщення роблять середовище непередбачуванішим і дозволяють перевірити, як політика поводиться з поведінкою, якої не було під час навчання. Третій сценарій також використовує випадкові стартові позиції, але є найскладнішим, зустрічний дрон намагається цілеспрямовано наблизитися до керованого дрона, постійно коригуючи свої координати *y* та *z*. Така «агресивна» тактика створює динамічну й потенційно небезпечну ситуацію, тестуючи, чи здатна модель зберегти безпечну дистанцію. Оскільки під час навчання модель бачила лише прямий політ із фіксованої точки, ці три сценарії показують, як різні алгоритми поводяться у відмінних умовах. Варіації стартових позицій, зигзагоподібні траєкторії та цілеспрямоване переслідування разом виявляють, наскільки адаптивними є стратегії натренованих моделей в умовах невизначеності.

У таблицях 1, 2 та 3 підсумовано результати роботи кожного алгоритму в трьох експериментальних сценаріях. У першому сценарії, де змінюються лише початкові координати зустрічного дрона, а його траєкторія лишається прямою, DQN демонструє високий показник успішності – 95 %, тоді як PPO досягає лише 70 %. DDPG і A2C показують середні результати – 71,75 % та 75,5 % відповідно. Найкраще спрацьовують SAC із 94,75 % та TD3, який уникає зіткнень у 100 % випадків. Це свідчить, що більшість алгоритмів справляються з невеликими зміщеннями стартової позиції, якщо траєкторія опонента залишається передбачуваною. Коли в другому сценарії пряма траєкторія замінюється зигзагоподібною, завдання стає значно складнішим. DQN не справився із завданням та не уникнув жодного зіткнення. Значення для PPO падає до 21,25 %, тоді як DDPG і A2C тримаються на рівні 40,75 % та 65,88 %. Натомість SAC зберігає майже той самий високий рівень, що й раніше 94,75 %, а TD3 знову демонструє бездоганний результат в 100 %. Такий розрив підтверджує вищу адаптивність TD3 та SAC порівняно з іншими методами. Третій сценарій є найскладнішим, зустрічний дрон активно зближається з натренованим дроном, постійно коригуючи координати y та z. DQN і PPO за цих умов зазнають невдачі у всіх спробах. A2C і DDPG дають змішані результати, DDPG успішний менш ніж у половині запусків, а A2C опускається нижче 20 %. Водночас SAC уникає зіткнень у 98,63 % випадків, а TD3 зберігає абсолютні 100 %. Порівняння SAC і TD3 показує, що обидва алгоритми здатні коригувати курс у польоті, хоча TD3 проходить повз опонента дещо прямолінійніше, а SAC залишає більшу дистанцію. Загалом, майже повне уникнення зіткнень доводить найвищий рівень адаптивності та узагальнювальної здатності цих двох методів щодо лінійного сценарію навчання.

На рисунках 2 і 3 показано траєкторії TD3 та SAC у третьому сценарії (проєкції *XY* і *XZ*). Оскільки координата *x* фіксована й задає прямолінійний рух уперед, основні відхилення спостерігаються у площинах *y* та *z*. Маркери часового кроку ілюструють швидкість зближення дронів і те, як навчений дрон маневрує, ухиляючись від небезпеки. Ці візуалізації підтверджують здатність SAC і TD3 протистояти дрону, що активно наближається. Загальні дані підтверджують, що TD3

Таблиця 1

Результати першого сценарію

Алгоритм	Успішні проходження	Зіткнення	Показник успішності
DQN	760	40	95 %
PPO	560	240	70 %
SAC	758	42	94.75 %
TD3	**800**	**0**	**100 %**
DDPG	574	226	71.75 %
A2C	604	196	75.5 %

Таблиця 2

Результати другого сценарію

Алгоритм	Успішні проходження	Зіткнення	Показник успішності
DQN	0	800	0 %
PPO	170	630	21.25 %
SAC	758	42	94.75 %
TD3	**800**	**0**	**100 %**
DDPG	326	474	40.75 %
A2C	527	273	65.88 %

Таблиця 3

Результати третього сценарію

Алгоритм	Успішні проходження	Зіткнення	Показник успішності
DQN	0	800	0 %
PPO	0	800	0 %
SAC	789	2	98.63 %
TD3	**800**	**0**	**100 %**
DDPG	351	449	43.88 %
A2C	151	649	18.88 %

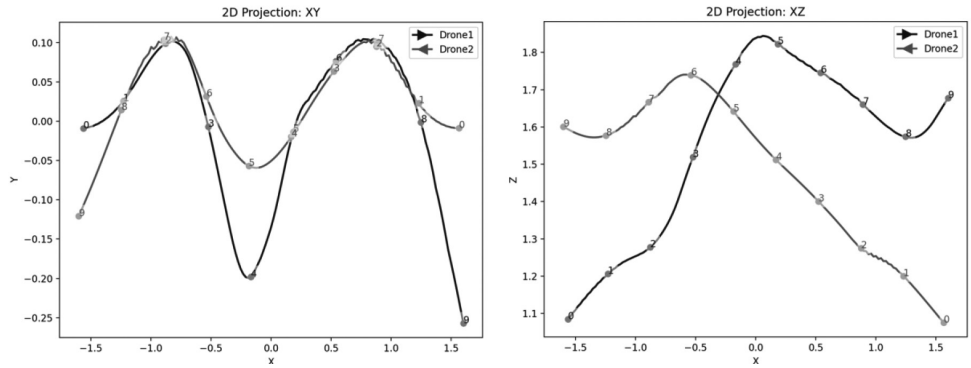

Рис. 2. Проєкції траєкторій TD3

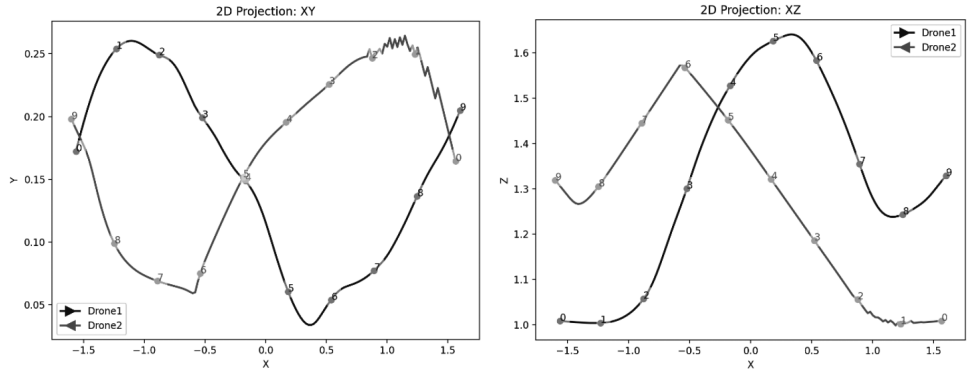

Рис. 3. Проєкції траєкторій SAC

і SAC найкраще справляються з непередбачуваними змінами траєкторії опонента, тоді як простіші алгоритми виявляють хороші результати лише в контрольованому середовищі та втрачають ефективність при значних відхиленнях.

Висновки. У цій роботі було досліджено, як різні алгоритми навчання, такі як DQN, PPO, SAC, TD3, DDPG та A2C, справляються зі завданням уникнення зіткнень БПЛА у тривимірному просторі. Контролюючи бокові та вертикальні компоненти швидкості одного дрона, що летить назустріч іншому, було створено сценарії, які перевіряли здатність кожного алгоритму адаптуватися до умов, відмінних від тренувальних. Оцінювання показало, що DQN і PPO добре працюють у простіших конфігураціях, але мають труднощі із складнішими траєкторіями опонента. Натомість SAC і TD3 майже бездоганно уникали зіткнень навіть у найскладніших умовах, що свідчить про їхню кращу здатність узагальнювати, коли рух зустрічного дрона суттєво відхиляється від початкової траєкторії навчання. Результати, отримані алгоритмами SAC і TD3 спонукають до подальших досліджень у реалістичніших умовах. Розширення симуляції кількома дронами, нелінійною аеродинамікою або складнішими сенсорними даними допоможе перевірити, чи зберігаються їхні адаптивні переваги. Відкритим залишається питання, наскільки добре натреновані моделі працюватимуть на фізичних платформах, оскільки чинники реального світу, зокрема вітер і перекриття огляду, тут не моделювалися.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ:

1. Mohsan S. A. H., Othman N. Q. H., Li Y., Alsharif M. H., Khan M. A. Unmanned aerial vehicles (UAVs): practical aspects, applications, open challenges, security issues, and future trends. *Intelligent Service Robotics*. 2023. Vol. 16, No. 1. P. 109–137. DOI: 10.1007/s11370-022-00452-4

2. Arulkumaran K., Deisenroth M. P., Brundage M., Bharath A. A. Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine*. 2017. Vol. 34, No. 6. P. 26–38. DOI: 10.1109/MSP.2017.2743240

3. Mnih V., Kavukcuoglu K., Silver D. *et al*. Playing Atari with deep reinforcement learning. arXiv Preprint. 2013. No. 1312.5602. URL: http://arxiv.org/abs/1312.5602 (дата звернення: 07.05.2025).

4. Schulman J., Wolski F., Dhariwal P. *et al*. Proximal policy optimization algorithms. arXiv Preprint. 2017. No. 1707.06347. URL: http://arxiv.org/abs/1707.06347 (дата звернення: 07.05.2025).

5. Hwang H. J., Jang J., Choi J. *et al*. Stepwise Soft Actor–Critic for UAV autonomous flight control. *Drones*. 2023. Vol. 7, No. 9. Article 549. DOI: 10.3390/drones7090549

6. Abo Mosali N., Shamsudin S. S., Alfandi O. *et al.* Twin delayed deep deterministic policy gradient-based target tracking for unmanned aerial vehicle with achievement rewarding and multistage training. *IEEE Access*. 2022. Vol. 10. P. 23545–23559. DOI: 10.1109/ACCESS.2022.3154388

7. Sun D., Gao D., Zheng J., Han P. Unmanned aerial vehicles control study using deep deterministic policy gradient. *2018 IEEE CSAA Guidance, Navigation and Control Conference (CGNCC)*. Xiamen, China. 2018. P. 1–5. DOI: 10.1109/GNCC42960.2018.9018682

8. Ayeelyan J., Lee G.-H., Hsu H.-C., Hsiung P.-A. Advantage Actor-Critic for autonomous intersection management. *Vehicles*. 2022. Vol. 4, No. 4. P. 1391–1412. DOI: 10.3390/vehicles4040073

9. Reuf K., Stefan W., Simon H. Deep Q-learning versus proximal policy optimization: Performance comparison in a material sorting task. *Proceedings of the IEEE International Symposium on Industrial Electronics (ISIE 2023)*. 2023. P. 1–6. DOI: 10.1109/ISIE51358.2023.10228056

10. Kalidas A. P., Joshua C. J., Md A. Q. *et al*. Deep reinforcement learning for vision-based navigation of UAVs in avoiding stationary and mobile obstacles. *Drones*. 2023. Vol. 7, No. 4. Article 245. DOI: 10.3390/drones7040245

11. Rybchak Z., Kopylets M. Comparative analysis of DQN and PPO algorithms in UAV obstacle avoidance 2D simulation. *Proceedings of the 8th International Conference on Computational Linguistics and Intelligent Systems. Volume III: Intelligent Systems Workshop*. 2024. P. 391–403. URL: https://ceur-ws.org/Vol-3688/paper25.pdf (дата звернення: 07.05.2025).

12. PyBullet physics engine. URL: https://pybullet.org/ (дата звернення: 07.05.2025).

13. Stable-Baselines3 documentation. URL: https://stable-baselines3.readthedocs.io/ (дата звернення: 07.05.2025).

14. Raffin A., Hill A., Gleave A. *et al*. Stable-Baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*. 2021. Vol. 22. Article 268. URL: http://jmlr.org/papers/v22/20-1364.html (дата звернення: 07.05.2025).

REFERENCES:

1. Mohsan, S. A. H., Othman, N. Q. H., Li, Y., Alsharif, M. H., & Khan, M. A. (2023). Unmanned aerial vehicles (UAVs): practical aspects, applications, open challenges, security issues, and future trends. *Intelligent service robotics, 16*(1), 109–137. https://doi.org/10.1007/s11370-022-00452-4

2. Arulkumaran, K., Deisenroth, M. P., Brundage, M., & Bharath, A. A. (2017). Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine, 34*(6), 26–38. https://doi.org/10.1109/MSP.2017.2743240

3. Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. A. (2013). *Playing Atari with deep reinforcement learning* [Preprint]. arXiv. http://arxiv.org/abs/1312.5602

4. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). *Proximal policy optimization algorithms* [Preprint]. arXiv. http://arxiv.org/abs/1707.06347

5. Hwang, H. J., Jang, J., Choi, J., Bae, J. H., Kim, S. H., & Kim, C. O. (2023). Stepwise Soft Actor–Critic for UAV autonomous flight control. *Drones*, *7*(9), 549. https://doi.org/10.3390/drones7090549

6. Abo Mosali, N., Shamsudin, S. S., Alfandi, O., Omar, R., & Al-Fadhali, N. (2022). Twin delayed deep deterministic policy gradient-based target tracking for unmanned aerial vehicle with achievement rewarding and multistage training. *IEEE Access, 10*, 23545–23559. https://doi.org/10.1109/ACCESS.2022.3154388

7. Sun, D., Gao, D., Zheng, J., & Han, P. (2018, August). Unmanned aerial vehicles control study using deep deterministic policy gradient. In *2018 IEEE CSAA Guidance,*

Navigation and Control Conference (CGNCC) (pp. 1–5). IEEE. https://doi.org/10.1109/GNCC42960.2018.9018682

8. Ayeelyan, J., Lee, G.-H., Hsu, H.-C., & Hsiung, P.-A. (2022). Advantage Actor-Critic for autonomous intersection management. *Vehicles, 4*(4), 1391–1412. https://doi.org/10.3390/vehicles4040073

9. Reuf, K., Stefan, W., & Simon, H. (2023). Deep Q-learning versus proximal policy optimization: Performance comparison in a material sorting task. In *Proceedings of the IEEE International Symposium on Industrial Electronics (ISIE 2023)* (pp. 1–6). IEEE. https://doi.org/10.1109/ISIE51358.2023.10228056

10. Kalidas, A. P., Joshua, C. J., Md, A. Q., Basheer, S., & Sakri, S. (2023). Deep reinforcement learning for vision-based navigation of UAVs in avoiding stationary and mobile obstacles. *Drones, 7*(4), Article 245. https://doi.org/10.3390/drones7040245

11. Rybchak, Z., & Kopylets, M. (2024). Comparative analysis of DQN and PPO algorithms in UAV obstacle avoidance 2D simulation. In *Proceedings of the 8th International Conference on Computational Linguistics and Intelligent Systems: Volume III, Intelligent Systems Workshop* (pp. 391–403). CEUR-WS. https://ceur-ws.org/Vol-3688/paper25.pdf

12. PyBullet. *PyBullet physics engine* [Computer software]. Retrieved from https://pybullet.org/

13. Stable-Baselines3. *Stable-Baselines3 documentation*. Retrieved from https://stable-baselines3.readthedocs.io/

14. Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M., & Dormann, N. (2021). Stable-Baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research, 22*, Article 268. http://jmlr.org/papers/v22/20-1364.html