INTELLECTUAL INFORMATION TECHNOLOGIES
ІНТЕЛЕКТУАЛЬНІ ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

https://doi.org/10.15407/intechsys.2025.03.030
УДК 004.8 + 004.032.26

О.А. УРСАТЬЄВ, канд. техн. наук, старш. наук. співроб., пров. наук. співроб.,
Інститут інформаційних технологій та систем НАН України,
просп. Акад. Глушкова, 40, м. Київ, 03187, Україна
https://org/0009-0009-8323-0525
aleksei@irtc.org.ua
О.Є. ВОЛКОВ, канд. техн. наук., старш. дослідник, директор,
Інститут інформаційних технологій та систем НАН України,
просп. Акад. Глушкова, 40, м. Київ, 03187, Україна
https://orcid.org/0000-0002-5418-6723
alexvolk@ukr.net

ПІДХОДИ ДО СТВОРЕННЯ МУЛЬТИАГЕНТНИХ СИСТЕМ І ГЛИБОКОГО ПОСИЛЕНОГО НАВЧАННЯ

Розглянуто зарубіжний досвід розробки та застосування штучного інтелекту за допомоги глибокого посиленого навчання нейромереж для розв'язання проблем з якими стикаються рухливі об'єкти у невідомих, можливо частково спостережуваних середовищах, для опису яких не існує математичної моделі. Надано таксономію різних завдань, що виникають при управлінні БПЛА чи роєм дронів, і наведено запропоновані безмодельні алгоритми глибокого посиленного навчання для розв'язання кожного з них. Виконано математичну формалізацію завдань у сфері управління БПЛА при посиленому навчанні, зокрема розглянуто парадигму навчання у багатоагентному середовищі. Розглянуто рішення деяких завдань щодо використання БПЛА.

Ключові слова: *безпілотні рухомі об'єкти, безпілотні літальні апарати БПЛА, управління роєм БПЛА, глибоке посилене навчання, ментальна модель світу, навчання нейромережі в уяві з застосуванням моделі світу, парадигми навчання агентів та схеми виконання завдань у багатоагентному середовищі.*

Цитування: Урсатьєв О.А., Волков О.Є. Підходи до створення мультіагентних систем і глибокого посиленого навчання дронів. *Information Technologies and Systems*, Київ, 2025, Том 3 (3), 30–55. https://doi.org/10.15407/intechsys.2025.03.030

Вступ

Безпілотні літальні апарати (БПЛА, *Unmanned Aerial Vehicles* (*UAVs*)) все частіше використовуються у багатьох складних та різноманітних завданнях, що належать до цивільних та військових галузей. БПЛА — це клас літальних апаратів, які можуть літати без присутності на борту людини-пілота. Їх зазвичай називають дронами. Вони можуть функціонувати з різним ступенем автономності: під дистанційним керуванням оператора-людини або автономно за допомоги бортових комп'ютерів. Вимога значного рівня автономності диктується необхідністю виконувати заплановані завдання у несподіваних ситуаціях без втручання людини. В огляді наведено стиснену класифікацію БПЛА і сформульовано мету, а саме: забезпечити бажані рівні досяжності автономності та функціонування БПЛА на кожному з етапів польотного завдання, та обгрунтовано стан використовуваних на той час методів рівня розвитку *state-of-the-art* глибокого посиленого навчання (навчання з підкріпленням, *Reinforcement Learning*, *DRL*) [1]. Наведені методи *DRL* були застосовані для забезпечення стабільної та плавної навігації БПЛА шляхом навчання *середовищ, змодельованих на комп'ютері*.

Класифікацію БПЛА було зроблено за багатьма важливими аспектами, такими як конфігурація верхнього рівня, яка передбачає виконання крила фіксованим, гвинтокрила та гібридного, гранична висота, середня злітна вага, рівень автономності тощо. Тип БПЛА із фіксованим крилом має жорстке крило з аеродинамічним профілем, що працює, збільшуючи поступальну швидкість польоту як звичайні літаки. Ці БПЛА підтримують тривалі польоти на витривалість та баражування, забезпечує високошвидкісний рух та підтримує високе корисне навантаження порівняно з конфігурацією гвинтокрилих БПЛА. Деякі з проблем полягають у необхідності злітно-посадкової смуги внаслідок вимоги поступальної швидкості повітря, завдання зависання не виконуються, оскільки ці БПЛА повинні продовжувати безперервний політ до посадки наприкінці будь-якого польоту.

Безпілотний вертоліт. Його лопасті, що обертаються, можуть створювати аеродинамічну силу тяги, достатню для польоту без необхідності відносної швидкості повітря. Ця конфігурація БПЛА має перевагу маневровості: може виконувати вертикальний зліт / посадку, літати на малих висотах, наприклад, у складних міських умовах, виконувати завдання зависання. Однак такий БПЛА не може зберігати те ж корисне навантаження, яке підтримує конфігурація з фіксованим крилом. Технічно ця конфігурація далі поділяється на підкатегорії залежно від кількості задіяних роторів; вона охоплює одноротоpні (вертольоти) і багатороторні (трироторні дрони, квадрокоптери тощо). Одноротоpні механічно складні і високовартісні, оскільки вони можуть злітати або приземлятися вертикально, підтримувати відносно високе корисне навантаження, тоді як багато-

роторні набагато швидші і здатні зависати або переміщатися навколо мети дуже плавно.

Гібридна конфігурація — це особливий тип повітряної платформи, що класифікуються як конвертоплани та хвостові літальні апарати. Гібридні БПЛА поєднують у собі переваги мультикоптерів та літаків з фіксованим крилом: вертикальний зліт, посадка та низьке енергоспоживання. Мають складну, змішану динаміку.

У 2005 році було введено рівні автономного управління (*Autonomous Control Levels, ACL*) або рівня незалежності від участі людини у пілотуванні. Кожен рівень спирається на три обставини: рівень незалежності, складність виконуваного завдання та рівень конфігурації середовища [1]. Перший — дистанційно пілотований. Цей рівень повністю контролює сертифікований експерт, який отримує візуальний зворотний зв'язок або сенсорні дані. Другий — дистанційно керований, званий напівавтономним. Цей рівень дозволяє транспортному засобу керувати собою на основі рішень, які приймаються під час спостереження за пілотом. Останній — повністю автономний рівень. На цьому рівні транспортному засобу надаються загальні завдання для виконання, і він стає здатним робити це, не уточнюючи, як виконати завдання.

У [1] розглянуто три основні проблеми, з якими стикаються БПЛА: планування шляху, навігація та керування (*path planning of drones, navigation, and control*). У складних системах, до яких, безумовно, відноситься БПЛА, із застосуванням штучного інтелекту (*AI*) для розв'язання цих проблем, наразі використовується посилене навчання (*RL*). Для забезпечення бажаного функціонування БПЛА було адаптовано безліч наявних та розроблені алгоритми *DRL* і згенероване на їх основі *AI*, що реалізує допомогу у невідомих середовищах, для опису яких немає математичної моделі.

Статтю присвячено аналізу сучасного зарубіжного досвіду застосування аналітичних платформ керування рухомими об'єктами, зокрема безпілотними автомобілями та БПЛА, у завданнях планування шляху, навігації та керування за допомогою штучного інтелекту, який генерується глибокими нейронними мережами, а саме посиленим навчанням у складному середовищі.

Мета статті — ознайомити фахівців з предметної області, чия основна робоча функція перебуває поза цариною машинного навчання, із проблемами, пов'язаними із застосуванням штучного інтелекту для розв'язання цих завдань, надійних та складних глибоких нейронних мереж а також їх навчання, що залишається складним завданням і потребує великої кількості даних та практичного досвіду і знань. Це може бути формою громадянської науки *(citizen science)* і сприятиме відтворенню досліджень та демократизації штучного інтелекту [2].

Огляд глибокого посиленого навчання (*Deep Reinforcement Learning, DRL*)

Навчання через взаємодію є основоположною ідеєю, що лежить в основі майже всього навчання та інтелекту. Найважливішою особливістю, яка відрізняє посилене навчання від інших типів навчання, є те, що воно використовує навчальну інформацію, яка оцінює виконані дії, а не інструктує, надаючи правильні дії. Глибоке навчання (*DL*) в нейронних мережах (*NN*) є актуальним для контрольованого навчання (*SL*) і самонавчання (*UL*) (рис. 1). Такі *NN* вчаться приймати / кодувати / передбачати / класифікувати шаблони або послідовності шаблонів, але вони не навчаються діяти в більш загальному сенсі посиленого навчання, *RL* в невідомих середовищах [3, 4].

Supervised and Unsupervised — контрольоване навчання і самонавчання відповідно. Самонавчання можна розділити на два підтипи: кластеризація та скорочення розмірності (*Dimension Reduction*). Кластеризація організовує дані в групи на основі критеріїв подібності. Скорочення розмірності займається відображенням даних із простору з високою розмірністю в простір з низькою розмірністю зі збереженням основних характеристик даних. *Reinforcement Learning* також можна розділити на два підтипи: Поведінка агентів та недиференційовані алгоритми, такі що не належать до жодного класу. Принцип навчання методом спроб і помилок при взаємодії агента з середовищем можна перефразувати для роботи з алгоритмами, які не можуть бути математично визначені. Їх можна навчити за допомоги посиленого навчання.

Посилене навчання (*RL*) — це парадигма для навчання задач прийняття рішень, яка може дати змогу агентам навчатися та адаптуватися до ситуацій у режимі онлайн. *RL* (див. рис. 1) — це форма навчання методом спроб і помилок, яка працює з винагородами та покараннями. Тобто, *RL* навчання — це навчання з того, що необхідно зробити і як зіставляти ситуації з діями, щоб максимізувати винагороду. Цей підтип навчання («Поведінка агентів», див. рис. 1)

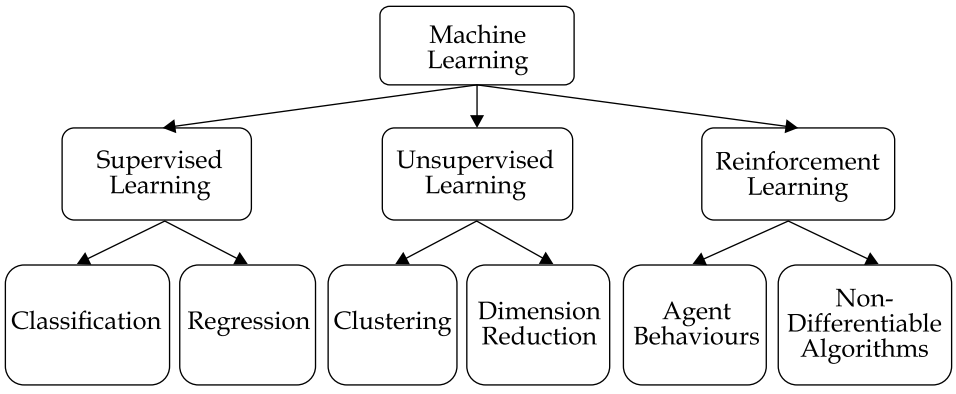

Рис. 1. Типи машинного навчання

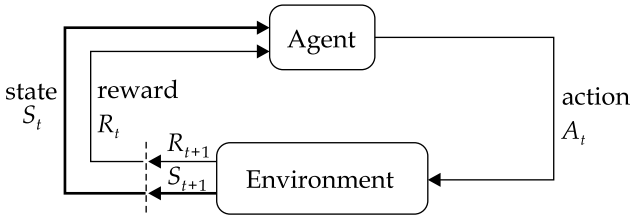

***Рис.* 2.** Взаємодія агента та середовища у Марківському процесі прийняття рішень [4, 7]

часто застосовується для агентних *симуляцій*[1], в яких агенту необхідно розробити стратегію успішної взаємодії з навколишнім середовищем. Рівень успіху вимірюється як винагорода, яку агент отримує від навколишнього середовища після виконання дії (рис. 2), що може бути отримана одразу або згодом. Одна з основних проблем *RL* стосується розуміння щодо присвоєння винагород за дії, які виконувалися в далекому минулому [5, 6].

Інформація, що циркулює між агентом та середовищем, у системі *RL*: *Environment* (Середовище), *Agent* (Агент), *State* (Стан), *Action* (Дія), *Reward* (Нагорода). Середовище — це світ, в якому діє агент. Середовище і агент в кожен момент часу мають певний стан. Агент може спостерігати за середовищем. Аналізуючи дані спостережень, агент вибирає дію, що призводить до зміни стану середовища. Після виконання дії, середовище реагує на нього, повертаючи агенту значення нагороди, та набуває нового стану.

Агент у контексті посиленого навчання — це сутність, яка інтерактивна зі середовищем, інтелектуальна, намагається досягти своєї мети і максимізувати свою нагороду, враховуючи поточний стан та досвід. Політика, винагорода, цінність та модель середовища є основними елементами *RL*. *Політика* визначає план дій агента (тобто, як агент реагує на різні ситуації навколишнього середовища та як він переводить стани в дії), або ймовірність того, що агент вдасться до певної дії, коли середовище перебуває в певному стані. *Винагороди* — це числові значення, які середовище дає агентові у відповідь на пару стан–дія. Ці значення винагороди описують негайну внутрішню бажаність станів навколишнього середовища. *Функція цінності* — це довгострокова версія функції винагороди, що обчислює дисконтовану дохідність, починаючи з певного стану, дотримуючись певної політики. Модель середовища надає поведінку середовища, що допомагає підвищити продуктивність алгоритму за допомоги розуміння довкілля. Основна мета будь-якого алгоритму посиленого навчання — дозволити агенту швидко вивчити оптимальну політику, яка точно досягає поставленого завдання і, таким чином, призводить

[1] Симулятор імітує керування, в загальному випадку, будь-яким процесом, і містить засоби, що відображають частину реальних явищ і властивостей у віртуальному середовищі, що створює враження дійсності.

до найбільшого значення винагороди. Глибоке посилене навчання готове зробити революцію в галузі штучного інтелекту і є кроком до створення автономних систем з більш високим рівнем розуміння візуального світу [1, 4—7].

Політика. Те, що агент намагається вивчити у сценарії *RL*, це політика, яка максимізує винагороду. Політика визначає ймовірність того, що агент вдасться до певної дії, коли середовище знаходиться в певному стані. Політика, яка максимізує загальну винагороду, називається оптимальною політикою. Вважається, що методи, засновані на політиці, більш стабільні, а методи, не засновані на політиці (*off-Policy*), більш ефективні та забезпечують кращий баланс між дослідженням нових політик та використанням найкращих на даний момент політик [3].

Підходи до *DRL* поділяють три основні категорії: засновані на цінностях (*Value-based approach*), на політиці (*Policy-based*), і підхід, заснований на моделі (*Model-based*) або без моделей (*Model-Free*). У навчанні на основі цінностей агент має на меті знайти політику, яка максимізує функцію цінності у довгостроковій перспективі у послідовності дій. Потім агент повинен знайти політику, що веде до оптимального значення для цільової функції. Ця категорія далі поділяється на детермінований та стохастичний підходи. Перша політика застосовує одну й ту саму дію у будь-якому стані, тоді як останній включає варіації дій, засновані на імовірнісних оцінках. Нарешті, *DRL* з моделлю залежить від надання агенту моделі середовища з вимогою до агента вивчити її для виконання завдань у цьому конкретному середовищі.

Незважаючи на відмінності, реалізовані в трьох підходах, вони мають деякі важливі характеристики, успадковані від концепцій *DRL*. Керування, реалізоване за допомоги *DRL*, діє як керування із замкнутим циклом, тоді як винагорода є зворотним зв'язком системи. Ця винагорода затримується, хоча багато алгоритмів намагаються зменшити цю затримку. Більше того, алгоритми, реалізовані в *DRL*, мають послідовну поведінку прийняття рішень, з довгостроковими винагородами, що залежать від послідовності дій. Концепція, яка називається проблемою призначення кредиту, описує залежність реалізації *DRL* від часу, оскільки деякі дії показують свої наслідки через деякий час і багато проміжних станів системи прийняття рішень [7].

На рис. 3 подано огляд типових та популярних алгоритмів посиленого навчання у структурному вигді [7, 8]. Алгоритми класифіковано з різних точок зору, включно з описаними раніше методами, та наведено їх комбінацію, засновану на цінностях та політиці, методи Монте-Карло (*MC*), тимчасової різниці (*TD*) та на основі політик і поза політикою.

***Model-Free* і *Model-Based* методи *RL*.** Існує два основних типи алгоритмів *RL*, які відрізняються один від одного тим, як вони працюють із ймовірностями переходу станів. Алгоритм *RL* з урахуванням

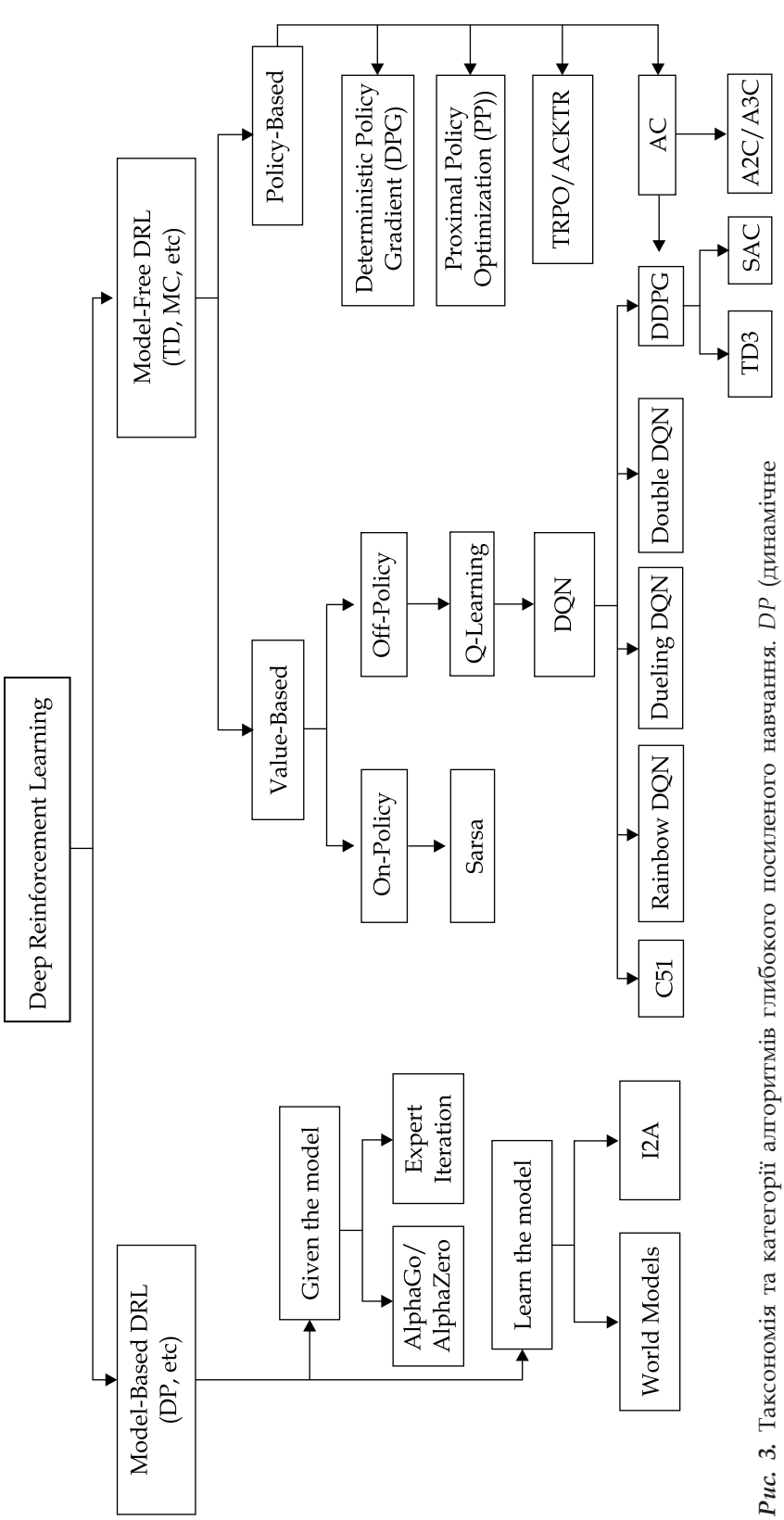

Рис. 3. Таксономія та категорії алгоритмів глибокого посиленого навчання. *DP* (динамічне програмування), *TD* (тимчасова різниця), *MC* (метод Монте-Карло), *I2A* (агент з розширеною уявою), *DQN* (глибока Q-мережа), *TRPO* (оптимізація політики довірчої області), *ACKTR* (актор-критик з використанням довірчої області з фактором Кронекера), *A3C* (асинхронний актор-критик з перевагою), *PPO* (оптимізація проксимальної політики), *DDPG* (глибокий детермінований градієнт політики), *TD3* (подвійний затриманий *DDPG*), *SAC* (м'який актор-критик)

моделей намагається моделювати ці можливості переходу, тоді як алгоритм *RL* без моделей цього не робить. Таким чином, алгоритм *RL* без моделей повністю слідує підходу проб і помилок, тоді як алгоритм *RL* на основі моделей робить це меншою мірою. Це загальна сутність всіх *Model-Free* алгоритмів у *RL*. Але саме вони досі є *state-of-the-art* (рівень розвитку: — найновіший, сучасний, тобто досягає найсучасніших результатів у цій задачі і є ефективнішим).

Навчання без моделі (*Model-Free*), лише на основі даних, що відбувається за допомоги механізму зворотного розповсюдження помилок, практично витіснило інші підходи в багатьох завданнях розпізнавання та оцінювання параметрів. Потребує багато навчальних прикладів. Перевагою є те, що оптимальні дії шукають градієнтним спуском. Недоліки. Основний з них — навчання з підкріпленням погано, а точніше, зовсім не працює з високими розмірностями. *Model-Free* методи можуть зійтися на ключових факторах, ігноруючи інші, але якщо алгоритму відразу не вдасться їх виявити, він швидше за все взагалі не навчиться. Також вони можуть зациклюватися на не оптимальній дії за умови, якщо до нього зійшовся градієнтний спуск, ігноруючи інші фактори. Навіть для незначно відмінних завдань нейромережу доводиться навчати заново.

Model-Based методи [5–6, 9–10] докорінно відрізняються від описаного підходу. *Model-Based* нейромережа тільки передбачає, що буде далі, не пропонуючи жодних дій, тобто є просто моделлю реальності (звідси «*Model-Based*»), а не системою прийняття рішень. *Model-Based* нейромережі легко навчаються внаслідок того, що вони просто пророкують як змінюватиметься середовище, не роблячи при цьому жодних пропозицій щодо оптимальності дій для збільшення нагороди. Нейромережа використовує для навчання всі наявні варіанти без винятку, а не тільки ті, що ведуть до зміни нагороди, як це відбувається у *Model-Free*. Разом з тим нейромережа повинна вивчити реальну динаміку системи, а отже, повинна бути достатньо ємною. Нейромережа у цьому разі є лише моделлю реальності, а оптимальні дії вибираються за допомоги зовнішнього планування завдань. Для здійснення необхідної дії необхідно змоделювати безліч випадкових дій і вибрати найкраще за цих обставин. Це класичний *Model-Based RL*. Однак за великих розмірностей і довгих ланцюжків, число можливих дій виходить занадто великим для перебирання. Тому *Model-Based* методи зазвичай поступаються *Model-Free*, які градієнтним спуском безпосередньо сходяться до найбільш оптимальних дій.

В [11] стверджується, що навчання на основі моделей зазвичай є ефективнішим з точки зору досвіду, оскільки для хорошого набування досвіду потрібно набагато менше затрат, а методи без моделей часто не приносять користі, оскільки час обчислень втрачається марно. За самостійного вивчення моделі середовища агент зіткнеться з деякими неточностями та невизначеністю, які можуть додатково вплинути на його політику та необхідні завдання. Таким чином, було

запропоновано низку підходів для інтеграції методів без моделей та на основі моделей [12]. Також і в [13] висловлено думку, згідно з якою у багатьох завданнях посиленого навчання [4] агент зі штучним інтелектом виграє від наявності хорошого уявлення минулих і дійсних станів, а також хорошої прогнозної *моделі майбутнього*[2,3], переважно потужної прогнозної моделі, такої як рекурентна нейронна мережа *RNN* [5, 6, 9]. Великі *RNN* — це дуже виразні моделі, які можуть вивчати багаті просторові та часові подання даних. Однак багато *Model-Free* методів *RL* змушені використовувати тільки малі нейронні мережі з невеликою кількістю параметрів. Алгоритм *RL* часто стикається з проблемою[4] присвоєння кредитів, через що традиційним алгоритмам *RL* важко вивчити мільйони ваги великої моделі, тому на практиці використовуються мережі меншого розміру, оскільки вони швидше переходять до хорошої політики під час навчання.

Оскільки вся система БПЛА заснована на взаємодії агента та середовища, досягнення оптимальної стратегії потребує високо розмірних вхідних даних та точного представлення середовища. Середовище, у свою чергу, може бути повністю або частково спостережуване. Це є двома основними парадигмами штучного інтелекту, кожна з яких має свої відмінні характеристики та виклики. Відсутність повної інформації створює труднощі у прийнятті рішень, оскільки агент може не повністю зрозуміти поточну ситуацію або вважатиме складним передбачити майбутні ситуації. Спираючись тільки на звичайні алгоритми *RL*, потрібні високоякісні та багатовимірні вхідні дані для створення точного опису станів БПЛА. Ці вимоги утворюють прогалину через можливості звичайного *RL*. Інтеграція між глибокою нейронною мережею та алгоритмами *RL* створена для заповнення цього пробілу [1] .

Ментальна модель світу. Наразі діє і все більше поширюється така концепція посиленого навчання — навчання нейромережі в уяві з використанням моделі світу [13]. У своїй доповіді *Recurrent World*

---

[2] *Werbos P.J., 1987. Learning How the World Works: Specifications for Predictive Networks in Robots and Brains. Proceedings of IEEE International Conference on Systems, Man and Cybernetics, N.Y.*

[3] *David Silver's Lecture on Integrating Learning and Planning [PDF] Silver D., 2017.*

[4] У багатьох завданнях *RL* зворотний зв'язок (позитивна чи негативна винагорода) дається в кінці послідовності кроків. Завдання присвоєння кредитів вирішує проблему з'ясування того, які кроки викликали результуючий зворотний зв'язок — які кроки мають отримати оцінку чи осуд за кінцевий результат. Керування, яке реалізується навчанням з підкріпленням, діє як замкнутий цикл, тоді як винагорода є зворотним зв'язком системи. Ця винагорода затримується, тоді як багато алгоритмів намагаються зменшити цю затримку. Більш того, алгоритми, реалізовані в *RL*, характеризуються послідовною поведінкою прийняття рішень із довгостроковими винагородами залежно від послідовності дій. Концепція, звана проблемою присвоєння кредиту, визначає залежність реалізації *RL* від часу, оскільки деякі дії виявляють свої наслідки через деякий час та безліч проміжних станів, пройдених системою [7].

Models Facilitate Policy Evolution (Рекурентні моделі світу сприяють еволюції політики навчання) на *NIPS* 2018 автори [13] запропонували рішення, яке поєднало в собі елементи навчання в уяві, що стали тепер класичними. Це дає змогу, застосовуючи універсальну модель світу, використовувати єдину *Model-Based* нейромережу для вирішення будь-якої кількості завдань. Модель світу — це система штучного інтелекту, яка створює внутрішнє уявлення середовища та використовує його для моделювання майбутніх подій у цьому середовищі [14]. Модель світу є ключем до інтелектуальних систем. Люди застосовують модель світу як симулятор у своєму мозку. Модель виходить за допомоги навчання на великих обсягах сенсомоторних даних у взаємодію з середовищем. Вивчити модель світу можна, використовуючи глибокі генеративні моделі. В основі вибору моделі навколишнього середовища — моделі світу — покладені відомі міркування. Люди розробляють ментальну модель світу, засновану на тому, що вони здатні сприймати своїми обмеженими органами чуття — це лише образ навколишнього світу, доступний нашому сприйняттю та опису з позицій вирішення поставленого завдання. Рішення та дії, які ми приймаємо і виконуємо, ґрунтуються на цій внутрішній моделі. Ніхто не уявляє собі весь світ у всьому його різноманітті. Вибираємо лише концепції та відношення між ними і використовуємо їх для подання реальної системи [13].

В [13] звернули увагу на те, щоб упоратися з величезною кількістю інформації, яка проходить через наше повсякденне життя. Мозок людини вивчає абстрактне подання як просторових так і тимчасових аспектів цієї інформації. Ми здатні спостерігати сцену та запам'ятовувати її абстрактний опис. Наявні дані також свідчать що те, що саме ми сприймаємо в будь-який момент, визначається пророкуванням нашим мозком такого майбутнього, яке засноване на нашій внутрішній моделі.

Один із способів зрозуміти прогностичну модель, яка існує всередині нашого мозку, полягає в тому, що вона може існувати не просто у пророкуванні майбутнього в цілому, а у пророкуванні майбутніх сенсорних даних з урахуванням наших поточних рухових дій. Ми здатні інстинктивно діяти відповідно до цієї прогностичної моделі і діяти швидко та рефлекторно, коли стикаємося з небезпекою, без потреби свідомо розгортати можливі сценарії майбутнього для формування плану відповідно до своїх передбачень майбутнього [13].

Математична формалізація завдань БПЛА за посиленого навчання

Multi-Agent RL (***MARL***). Багато мультиагентних систем природним чином містять кілька агентів, що приймають рішення, і які взаємодіють одночасно. Завдяки досягненням одноагентного глибокого *RL*, посилене багатоагентне навчання перейшло до багатоагентного

посиленого глибокого навчання (*MADRL*) з реальною складністю [15]. Посилене навчання формалізовано у Марківських процесах прийняття рішень (*MDP*) як основу для навчання з одним агентом, і у разі багатоагентного навчання — Марківську гру [15]. Традиційне завдання посиленого навчання [16] пов'язане з навчанням політики управління, що оптимізує числову продуктивність шляхом ухвалення рішень поетапно. Адаптивний агент, що приймає рішення, взаємодіє з середовищем невідомої динаміки методом проб і помилок з метою поліпшити свою роботу, що визначається імовірнісною функцією переходу. Стандартним формулюванням для такого послідовного ухвалення рішень є *MDP*. Взаємодія адаптивного агента із середовищем визначається імовірнісною функцією переходу станів. У цьому соліпсичному[5] поданні вторинні агенти можуть бути лише частиною середовища і, отже, є фіксованими у своїй поведінці. Структура Марківських ігор дозволила розширити це уявлення, долучивши до нього кілька адаптивних агентів із взаємодіючими чи конкуруючими цілями [17]. Тому, коли послідовне прийняття рішень поширюється на кількох агентів, як структуру зазвичай застосовують Марківські ігри[6]. Подібно до проблеми з одним агентом, мета кожного агента полягає в тому, щоб змінити свою політику таким чином, щоб оптимізувати отримані винагороди в довгостроковій перспективі. В одноагентному формалізмі агент є єдиним екземпляром, що приймає рішення, який впливає на стан середовища. Переходи станів можна чітко приписати агенту, тоді як усе, що знаходиться за межами поля впливу агента, сприймається як частина базової динаміки системи. Незважаючи на те, що середовище може бути стохастичним, проблема навчання залишається стаціонарною. Навпаки, одна з фундаментальних проблем у багатоагентній галузі полягає в тому, що агенти оновлюють свої політики під час процесу навчання одночасно, тому середовище здається нестаціонарним з точки зору одного агента. Отже, припущення Маркова про *MDP* більше не виконується, і агенти мають справу з проблемою цілі, яка рухається.

Також є зауваження щодо середовища. Відсутність повної інформації про середовище створює труднощі у прийнятті рішень, оскільки агент може не зрозуміти поточну ситуацію або вважатиме складним передбачити майбутні ситуації. Частковість спостереження може бути наслідком різних причин, таких як неточні датчики, обмежений діапазон датчиків або складність середовища.

[5] **Соліпси́зм** (лат. *solus ipse* — тільки сам) — філософська доктрина і позиція, крізь призму якої визнається власна індивідуальна свідомість як єдина і безперечна реальність навколишнього світу. Розглядається як крайня форма суб'єктивного ідеалізму. Трактується як заперечення реальності всього, крім власної свідомості. «*Мій розум — єдина річ, яка існує*.»

[6] Марківські ігри також відомі як стохастичні ігри (*Shapley* 1953). Термін «гра Маркова» використовується, щоб провести чітку різницю між детермінованими та стохастичними Марківськими іграми.

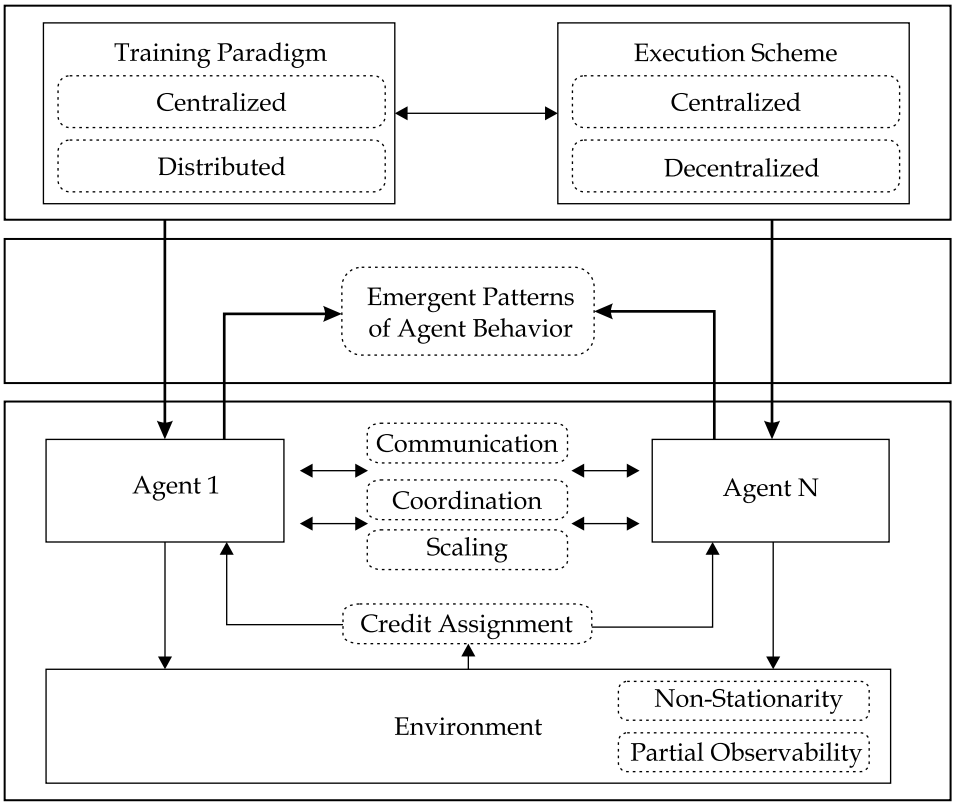

Рис. 4. Схематична структура матеріалу, викладеного в роботі [15]. Парадигми навчання (*Training Paradigms*) поведінки агентів у багатоагентному середовищі: централізована (*Centralized*) та розподілена (*Distributed*). Схема виконання (*Execution sheme*): централізована (*Centralized*) та децентралізована (*Decentralized*), коли агенти визначають дії відповідно до своєї індивідуальної політики. *Емерджентні моделі* поведінки агента (*Emergent Patterns of Agent Behavior*), обумовлені взаємодією сутностей, можуть викликати системну поведінку, яка відноситься до несподіваних та нелегко передбачуваних результатів. Таким збудником можуть бути структури винагороди, мови та соціального контексту.

У частково спостережуваному середовищі агенти повинні використовувати стратегії для вирішення проблеми частковості спостереження, включаючи оцінку стану, імовірнісні міркування та використання пам'яті. Формальні структури, такі як частково спостережувані процеси прийняття рішень Маркова (*Partially Observable Markov Decision Processes*, *POMDPs*), зазвичай використовуються для моделювання та вирішення проблем у таких середовищах, що дозволяє агентам розробляти складні стратегії, які збалансовують розвідку та розроблення [18].

В [15] подано критичний огляд поточних розробок в галузі *MADRL* з аналізом структури схем навчання для декількох агентів, а саме закономірності поведінки агентів у кооперативних, конкурентних та змішаних сценаріях. Систематизовано проблеми, які виникають виключно в багатоагентній сфері глибокого *RL* (рис. 4.),

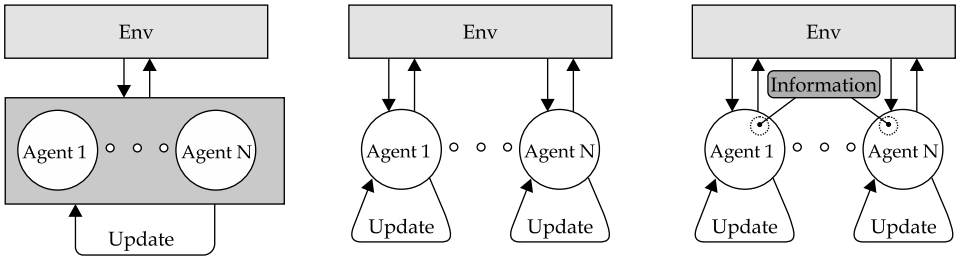

Рис. 5. Схеми навчання у багатоагентному середовищі. Зліва направо: *CTCE* підтримує загальну політику всім агентам; *DTDE* — кожен агент оновлює власну індивідуальну політику; *CTDE* дає змогу агентам обмінюватися під час навчання додатковою інформацією, яка потім відкидається під час виконання

та розглянуто сучасні рішення, які були запропоновані для розв'язання проблем. Обговорено досягнення, визначено тенденції та позначено можливі напрями робіт у цій галузі досліджень.

Парадигма навчання у багатоагентному середовищі. За централізованого навчання агентів *CTCE* (див. рис. 4) політики оновлюються з урахуванням взаємного обміну інформацією під час навчання. Навчання у розподілений спосіб *DTDE* відбувається у разі, коли кожен агент виконує оновлення самостійно та навчається індивідуальній політиці без використання зовнішньої інформації. Централізоване виконання передбачає, що всім агентам обчислюються спільні дії. Навпаки, агенти визначають дії відповідно до своєї індивідуальної політики для децентралізованого виконання *CTDE*.

Поточні проблеми *MADRL* охоплюють нестаціонарність середовища, що виникає через спільне адаптування агентів; навчання комунікації; необхідність узгодженої координації дій; проблему присвоєння кредиту; можливість масштабування до довільної кількості агентів, які приймають рішення; та неМарківські середовища через частковість їхніх спостережень (*Partial Observabilety*).

За централізованого навчання агентів *CTCE* (рис. 4) політики оновлюються з урахуванням взаємного обміну інформацією під час навчання. Навчання розподіленим чином *DTDE* відбувається у разі, коли кожен агент виконує оновлення самостійно та навчається індивідуальній політиці без використання зовнішньої інформації. Централізоване виконання передбачає, що агенти керуються з централізованого блоку, який обчислює спільні дії всім агентам. Навпаки, агенти визначають дії відповідно до своєї індивідуальної політики для децентралізованого виконання *CTDE*. Огляд схем навчання у багатоагентному середовищі зображено на рис. 5.

Парадигма централізованого навчання із децентралізованим виконанням[7] в одному і тому ж середовищі (*Centralized Training with*

[7] *MathWorks. MATLAB Answers. Centralized vs Decentralized Training for Multi Agent Reinforcement Learning. Commented: Lin on 22 Jul 2024. https://uk.mathworks.com/*

Decentralized Execution; *CTDE*). *CTDE* — це домінуючий підхід як для кооперативного, так і для змішаного середовища, завдяки своїй здатності ефективно навчати децентралізованої політики, є комбінацією незалежного і централізованого *MARL* [15]. У той час як у змішаних середовищах повна автономія агентів може бути бажаним результатом, кооперативне середовище дозволяє агентам обмінюватися інформацією для полегшення координації.

Повністю кооперативна обстановка означає, що агенти отримують однакову винагороду за переходи станів. У такій рівноправній обстановці винагороди агенти мотивовані до співпраці та намагаються уникнути невдачі окремого агента, щоб максимізувати продуктивність команди. У більш загальному плані йдеться про кооперативні умови, коли агенти заохочуються до співпраці, але не мають рівноправної нагороди.

Встановлено, що розглянуті алгоритми багатоагентного навчання *RL* показали, що основна складність полягає в узагальненні на безперервні дії та простір станів, а також масштабуванні на велику кількість агентів [15]. Навчання кількох агентів тривалий час було обчислюваним завданням. Оскільки складність у просторі станів та дій зростає експоненційно з кількістю агентів, навіть сучасні підходи глибокого навчання можуть досягати своїх меж. Визначено такі проблеми координації незалежних агентів, що навчаються, які виникають у повністю кооперативних Марківських іграх, зокрема нестаціонарність, стохастичність та тіньова рівновага [15].

Посилене навчання для керування БПЛА (*Reinforcement Learning for UAV Control*)

Керування БПЛА, окрім завдань навігації та планування шляху, як і раніше є широкою концепцією, яка охоплює безліч різних завдань починаючи від керування орієнтацією одного БПЛА до керування каналами зв'язку між декількома БПЛА. Також глибоке посилене навчання сприяє вирішенню нових складних завдань у цієї сфері, таких як безпечне керування гібридним БПЛА та керування роєм дронів (*UAV Swarm control or swarm of UAVs*). Ця сфера керування до виконання певних завдань з мінімальними ресурсами (мінімальними переміщеннями, часом, енергією тощо) дуже складна. Складність полягає у самій її природі: складна, змішана динаміка, взаємодія з експлуатаційними та екологічними умовами, що змінюються, потребою бути енергетично ефективною та її вразливістю до перешкод, шумів датчиків і не модельованої динаміки. Поза тим, існує багато завдань з помітними варіаціями, в яких потрібне стабільне та точне керування БПЛА. На рис. 6 надано таксономію різних завдань, що

matlabcentral/answers/2002007-centralized-vs-decentralized-training-for-multi-agent-reinforcement-learning

виникають в разі керування БПЛА, і вказано запропоновані *Model-Free DRL* алгоритми для кожного з них — рис. 1 [1].

У цьому контексті *DRL* було протестовано в режимі реального часу із застосуванням різноманітних *Model-Free* алгоритмів на основі базових політик, таких як, *DDPG*[8], *PPO*, *TRPO*[9] тощо (рис. 3). Кожен алгоритм показав свої переваги та недоліки, що робить їх легкими для одних застосувань і незручними для інших [1].

Основною темою були реальні застосування БПЛА, які використовуються для утримання дронів у бажаному положенні та орієнтації для виконання певного завдання. Йдеться про формування зовнішнього контуру, що утворюється роєм дронів, які можуть підтримувати комунікації між собою. Враховуючи складність завдання, високу швидкість переміщення дронів та точність, яка має бути гарантована для керування роєм, застосовувалося посилене глибоке навчання. Було застосовано декілька підходів, та дослідження у цій галузі ще не завершено [1]. Наприклад, політика керування глибокою згортковою нейронної мережею (*DCNN*[10] *P-policy*) дала змогу керування п'ятьма БПЛА [19]. Завдання роя полягали в тому, щоб автономно вишикуватися у певну форму та захистити територію від нападників. Мотивоване необхідністю кооперативної поведінки дронів, багатоагентне посилене навчання, *MARL* застосовувалося для вирішення проблеми відсутності інформації про всі змінні стани для всіх БПЛА в рої [15]. У цій проблемі керування в алгоритмі *DCNNP* застосовувалися централізовані підходи до навчання та централізованого виконання. Тільки одному БПЛА, названому агентом, було дозволено вивчати модель, планувати поведінку для всіх інших БПЛА та спілкуватися з ними про їхні завдання. Поведінка керуван-

[8] ***DDPG*** (*Deep Deterministic Policy Gradients*) — це алгоритм, що поєднує у собі ідеї з двох сфер: *DPG* (*Deterministic Policy Gradients*) та *DQN* (*Deep Q-Network*), алгоритм посиленого навчання, який поєднує в собі елементи *Q*-навчання (*Q*-функція оцінки довгострокової нагороди) та методу актор-критик (*Actor-Critic*). У *DDPG* використовується нейронна мережа для параметризації політики (актор) та критика (критика) для оцінки *Q*-функції стану-дії. Підходить для завдань з безперервною дією, став основою, наприклад, для керування роботами та автономного водіння.

PPO (*Proximal Policy Optimization*) — це алгоритм посиленого навчання, який використовує методи оптимізації політики, щоб навчати агентів приймати дії у середовищі. У ньому використовується актор-критик, де актор (політика) оновлює поліпшення вибору дій, а критик (оцінювач) оцінює, наскільки добре дії агента відповідають очікуваним нагородам. *PPO* є популярним вибором завдяки своїй стабільності та хорошій продуктивності в різних середовищах — *https://habr.com/ru/companies/otus/articles/771412/*

[9] ***TRPO*** (*Trust Region Policy Optimization,* Оптимізація політики довірчої області) — це алгоритм посиленого навчання, який грунтується на методі максимізації функції корисності з обмеженням (*trust region*). Відомий своєю здатністю забезпечувати стабільне навчання та гарантувати покращення політики.

[10] *DCNN* — Глибока згорткова нейронна мережа (*CNN*) з кількома шарами, яка зазвичай використовується для аналізу та розпізнавання зображень — *https://builtin.com/articles/dcnn*

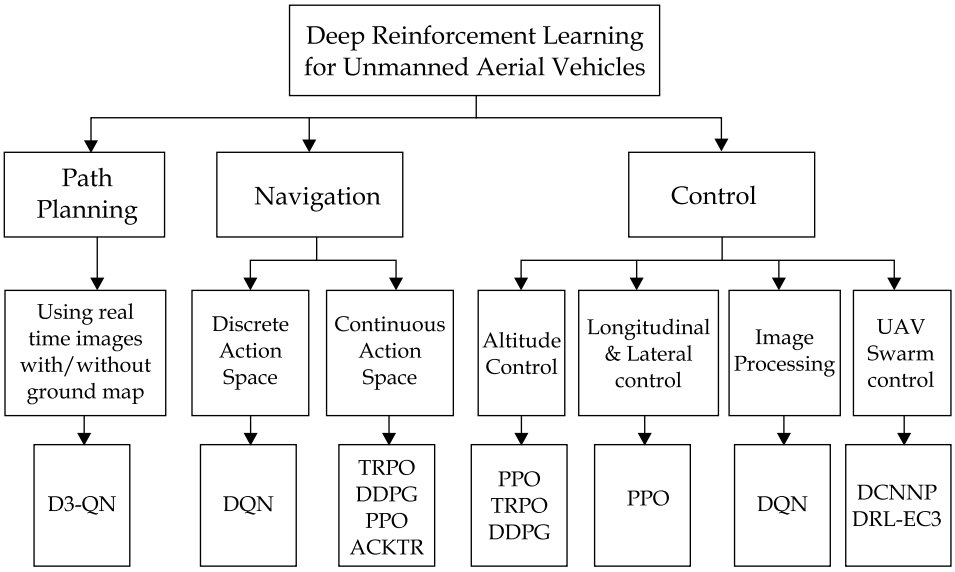

Рис. 6. Таксономія алгоритмів глибокого посиленого навчання у завданнях БПЛА, що наведені у третьому зверху ряду: *Using real time images with / without ground map* — використання зображень у реальному часі з картою місцевості / без неї, *Discrete or Continuous Action Space* — дискретний або безперервний простір дій, *Altitude control* — контроль висоти, *Longitudinal & Lateral control* — поздовжній і бічний контроль, *Image Processinng* — обробка зображень; *UAV Swarm control* — керування роєм БПЛА

Algorithm	Agent Type	Policy	Policy Type	MC or TD	Action Space	State Space
State action reward state action (SARSA) SARSA Lambda	Value-based	On-policy	Pseudo-deterministic (ϵ – *greedy*)	TD	Discrete only	Discrete only
Deep Q Network (DQN) Double DQN Noisy DQN Prioritized Replay DQN Dueling DQN Categorical DQN Disturbuted DQN (C51)	Value-based	Off-policy	Pseudo-deterministic (ϵ – *greedy*)		Discrete only	Discrete or Continuous
Normalized Advantage Functions (NAF) = Continuous DQN	Value-based				Continuous	Continuous
REINFORCE (Vanilla policy gradient)	Policy-based	On-policy	Stochastic	MC		
Policy Gradient	Policy-based		Stochastic			
TRPO	Actor-critic	On-policy	Stochastic		Discrete or Continuous	Discrete or Continuous
PPO	Actor-critic	On-policy	Stochastic		Discrete or Continuous	Discrete or Continuous
A2C/A3C	Actor-critic	On-polciy	Stochastic	TD	Discrete or Continuous	Discrete or Continuous
DDPG	Actor-critic	Off-policy	Deterministic		Continuous	Discrete or Continuous
TD3	Actor-critic				Continuous	Discrete or Continuous
SAC	Actor-critic	Off-policy			Continuous	Discrete or Continuous
ACER	Actor-critic				Discrete	Discrete or Continuous
ACKTR	Actor-critic				Discrete or Continuous	Discrete or Continuous

Рис. 7. Зведена таблиця безмодельних алгоритмів *RL* у завданнях БПЛА

ня *DCNNP* порівнювалася з поведінкою імовірнісної політики, поданої випадковим переміщенням БПЛА до досягнення місії, та ідеальної / периметральної політики, що досягається рівномірним переміщенням БПЛА в одному напрямку. Результати показали, що *DCNNP* значно перевершив випадкову політику обох задач; спостерігалися невеликі відмінності у продуктивності порівняно з підходом периметра для оборони об'єкта, але зі зниженою продуктивністю порівняно з ідеальною політикою, особливо коли кількість БПЛА невелика у задачі спостереження.

Інше дослідження, пов'язане з керуванням роєм, було запропоновано в [20]. БПЛА можуть застосовуватися як повітряні базові станції для покращення покриття та продуктивності комунікаційних мереж, зокрема як екстрений зв'язок та доступ до мережі для віддалених районів. Вони можуть встановлювати канали зв'язку для наземних користувачів у разі доставки пакетів. Однак БПЛА мають обмежені радіус дії зв'язку та енергетичні ресурси. Зокрема, для великого регіону вони не можуть постійно покривати всю територію або продовжувати політ протягом тривалого часу. Таким чином, складно керувати групою БПЛА для досягнення певного покриття зв'язку в довгостроковій перспективі, зберігаючи при цьому їх зв'язок та мінімізуючи споживання енергії. З цією метою було запропоновано енергоефективне керування БПЛА на основі *DRL* за новими методом і алгоритмом, що створено, який здійснює підключення до зони покриття мережі (*DRL Based Energy Efficient Control for Coverage Connectivity* (*DRL – EC3*)) [20]. Він застосовує метод глибокого детермінованого градієнта політики (*Deep Deterministic Policy Gradient* (*DDPG*)) з деякими модифікаціями та політикою актор-критик, що використовує двошарову глибоку нейронну мережу для кожного з них. Цей алгоритм реалізовано на *TensorFlow*[11] з 400 нейронами в першому шарі та 300 у другому в кожній мережі. Основною метою цього дослідження була економія енергоспоживання роєвих БПЛА під час роботи та зв'язку без зниження точності виконання завдань. Винагорода, що використовувалася в навчанні, ґрунтувалася на покритті, індексі справедливості покриття, споживанні енергії та безперервному зв'язку для всіх БПЛА. Це дослідження показало, що запропонована модель керування *DRL-EC3* перевершує як відомі випадкові, так і жадібні (*Greedy algorithm*[12]) політики. Понад те, дослідження

[11] *TensorFlow* — відкрита програмна бібліотека для машинного навчання, розроблена *Google* для вирішення завдань побудови та тренування нейронної мережі, дозволяє швидко створювати нейромережі.

[12] Жадібні алгоритми — це клас алгоритмів, які роблять локально оптимальний вибір на кожному кроці з надією знайти глобально оптимальне рішення, тобто алгоритм є простим і скоріше евристичним, який приймає найкраще рішення, виходячи з наявних на кожному етапі даних. Легкий в реалізації і часто дуже ефективний за часом виконання, але багато задач не можуть бути розв'язані за його допомоги — *https://www.geeksforgeeks.org/greedy-algorithms/*

показало, що зі збільшенням кількості БПЛА новий алгоритм керування виявився успішним [20].

Ще одне дослідження вирішує проблему роєм (кластером) БПЛА для виконання планування завдань у реальному часі [21]. Останнім часом рої БПЛА широко застосовуються завдяки їхній високій гнучкості, широкому охопленню зони покриття зв'язку та надійній ефективності передачі інформації. Для досягнення співпраці БПЛА в разі виконання кількох завдань кластеру запропоновано алгоритм планування завдань на основі посиленого глибокого навчання, який дає змогу БПЛА автоматично і динамічно коригувати свою стратегію виконання завдань, використовуючи власні обчислення ефективності їх виконання. Оскільки БПЛА необхідно виконувати завдання, працюючи в динамічному середовищі без централізованого керування, йому необхідно навчатися завданням відповідно до даних теж у реальному часі. Посилене навчання здатне навчатися та приймати рішення в реальному часі на основі інформації у середовищі, що є доречним та здійсненним методом планування завдань кластерів БПЛА.

З цієї точки зору, розглядалось посилене навчання, яке вирішує проблему розподілу каналів, що існує в плануванні завдань кластерів БПЛА. Цей підхід полягає у виділенні каналів для підтримки зв'язку для завдань керування БПЛА, таких як оцінювання стану та відстеження траєкторії. Таке завдання передбачає величезну кількість даних, отриманих в результаті навчання, тренування та обробки цих даних у реальному часі. Враховуючи це, *Q*-навчання було виключене з варіантів через його складність оброблення великих наборів даних. Натомість використовувалася глибока мережа посиленого навчання через її здатність подавати дані в нижчій розмірності та отримувати значення *Q* більш ефективно. Архітектуру керування можна описати як багатоагентну з децентралізованим керуванням, при цьому не потрібні ніякі моделі для системи, а поведінка агентів розглядається з погляду кожного агента як спостереження за станом навколишнього середовища. Алгоритм глибокого посиленого навчання розглядає чотири випадки стану каналу. По-перше, якщо канал зайнятий, система чекає, доки дані не будуть надіслані знову. По-друге, якщо передача не вдалася, дані мають бути надіслані повторно. По-третє, якщо передача успішна, наступні завдання можуть бути виконані. По-четверте, якщо немає завдання, ліниво чекайте. Оскільки головна мета полягає в якості передачі, винагорода заснована на максимальній кількості переданих даних та меншій затримці передачі. Запропонована стратегія успішно забезпечує точну та швидку робочу поведінку для зв'язку між БПЛА [1].

Порівняльна оцінка застосованих алгоритмів RL у вирішені завдань БПЛА

Складна проблема реального застосування дронів потребує вирішення невизначеностей та адаптації до динамічних невідомих середовищ. *DRL* пропонує гнучку основу для вирішення цих

проблем без застосування моделі (*Model-Free*) та може навчатися на досвіді й передбачати правильне рішення для майбутніх спостережень. Із двох гілок алгоритмів *Model-Based RL* намагаються вибрати оптимальну політику на основі вивченої моделі середовища, тоді як у безмодельних алгоритмах оптимальна політика вибирається на основі помилок, з якими стикається агент (див. рис. 7). Стратегія керування агентом (функція прийняття рішень), яка є відображенням ситуацій у дії, називається політикою, й існує її два типи, тобто політика відповідності та політика поза політикою. Методи, що базуються на політиці (*SARSA*, *PPO*, *TRPO*), намагаються оцінити або покращити політику, яка використовується для прийняття рішень, тоді як методи, що не базуються на політиці (*Q*-навчання, *DQN*, *DDPG*), оцінюють або покращують політику, відмінну від тієї, що використовується для генерації даних [1].

Усі алгоритми досягають певного стаціонарного стану, однак, *TRPO* та *DDPG* дають екстремальні коливання як по осі крену, так і по осі рискання, що може спричинити нестабільність під час польоту. Однак, загалом кажучи, з точки зору похибки, *PPO* показав себе як більш точний контролер положення завдяки його методу градієнта політики першого порядку, який особливо популярний завдяки чудовим характеристикам у тестах та простоті реалізації. Для навчання завдань керування польотом дрона проведено експерименти з використанням *PPO*, *DDPG* та *TRPO* для зависання, посадки, випадкових точок маршруту та супроводження цілі [1]. Виявлено, що *PPO* загалом був найбільш послідовним алгоритмом для посадки, за ним йшов *TRPO*. Для зависання *DDPG* забезпечував дуже плавний політ порівняно з іншими алгоритмами. У завданні випадкової навігації за точкою маршруту *PPO* забезпечував найплавнішу навігацію порівняно з іншими алгоритмами. Для завдання слідування за ціллю *DDPG* забезпечує кращі середні винагороди та найплавнішу навігацію за точкою маршруту, за ним йшли *PPO* та *RTPO*.

Для великих складних невідомих середовищ деякі ідеї спрямовано на застосування методів *DRL* для просування до своїх завдань, покладаючись тільки на сенсорні дані і сигнали *GPS*. Не має необхідності планувати невизначену 3*D*-модель для невідомого середовища та планувати шляхи для проходження транспортних засобів, так стверджують автори, — БПЛА зможе розумно переміщатися з довільних місць відправлення у довільні цільові позиції [22]. В цій роботі, пов'язаній з доставою вантажів за допомоги БПЛА, що нині актуально, моделюється автономна навігація БПЛА у масштабному невідомому складному середовищі як безперервне завдання керування у дискретному часі, та вирішується ця задача за допомоги *DRL*. Без планування шляху або побудови карти пропозицій запропонований метод дозволяє БПЛА переміщатися до пунктів призначення, використовуючи тільки сенсорну інформацію локального середовища та сигнал *GPS*. Прийнято, що завдання навігації є

Марківським процесом, який частково спостерігається, прийняттям рішень (*POMDP*), а також зауважуємо, що рекурентний детермінований алгоритм градієнта політики менш ефективний. Отже, створено швидший алгоритм навчання політиці для *POMDP* з урахуванням архітектури актор-критик. Для перевірки змодельовано п'ять віртуальних середовищ та віртуальний БПЛА, який летить на фіксованій висоті з постійною швидкістю. Пізнання локального середовища досягається шляхом вимірювання відстаней від БПЛА до перешкод у кількох напрямках. Результати моделювання показали ефективність такого підходу.

До слова, рішення для уникнення об'єктів може бути досягнуто лише шляхом зміни висоти БПЛА без переміщення праворуч або ліворуч. Застосований метод *DRL* вивчає значення *Q* і оптимальну політику для уникнення об'єктів. Можна також застосовувати інші політики та методи, такі як асинхронний актор-критик з перевагою (*A3C*), глибокий детермінований градієнт політики (*DDPG*) та архітектуру дуельної мережі для мереж *Q* з подвійною глибиною (*D3QN*)[13]. Варто лише змінити функцію втрат [1], [23].

Модний алгоритм *DQN* страждає від суттєвого переоцінювання дії-стану в задачах посиленого навчання, таких як ігри в домені *Atari* 2600 та домені планування шляху. Щоб зменшити переоцінювання значень дій під час навчання, представлено нову комбінацію алгоритму подвійного *Q*-навчання та дуельного *DQN*, а також розроблено алгоритм під назвою «Варіант подвійного дуельного *DQN*» (*V-D D3QN*) [23]. Автори зосередились на ідеї, що лежить в основі алгоритму *V-D D3QN*, та запропонували можливу ідею застосування двох дуельних *DQN*-мереж для зменшення переоцінки значень дій під час навчання. Конкретний підхід полягає у випадковому виборі однієї дуельної *DQN*-мережі на кожному кроці часу для оновлення її параметрів, використовуючи решту дуельної *DQN*-мережі для визначення цілей оновлення. Потім здійснили експерименти в налаштованому віртуальному середовищі-мережі. Результати експериментів показали, що запропонований алгоритм не тільки зменшує переоцінки ефективніше, ніж алгоритм *Double DQN* (*DDQN*), але й призводить до набагато кращої продуктивності в області планування маршруту з великою здатністю до узагальнення нових та швидкозмінних середовищ [23].

Що стосується загального поліпшення систем, слід взяти до уваги, що умови навколишнього середовища, такі як швидкість вітру, дощі та пил, вносять невизначеність в результати. Тому їх слід розглядати як системні перешкоди і потім професійно обробляти [1].

[13] Архітектура дуелі складається з двох потоків, які представляють функції цінності та переваги, при цьому розділяючи загальний модуль навчання згорткових ознак. Два потоки об'єднуються за допомоги спеціального шару, що агрегує, для отримання оцінки функції цінності стану-дії *Q* [23].

Навігація та керування БПЛА стали новою сферою через широкий спектр їх застосувань. БПЛА використовують в різних аспектах життя, таких як цивільні завдання, військові місії, відстеження об'єктів та пошуково-рятувальні операції [24]. Кожна з цих програм вимагає точної навігації, щоб уникати зіткнень під час вибору оптимального маршруту для досягнення мети. Навігація в невідомому середовищі з перешкодами, що змінюються, є однією з проблем керування БПЛА [1]. Для вирішення цієї проблеми було розроблено та оцінено різні алгоритми керування та методи штучного інтелекту.

DRL є одним з перспективних рішень навігації БПЛА. Тоді як іншим алгоритмам машинного навчання потрібні марковані дані для початку процесу навчання, що недоступно у випадку з БПЛА, оскільки вони мають справу з новими середовищами під час кожної місії [22]. Використання *RL* у навігації БПЛА переважно спирається на функцію винагороди, що визначається діями БПЛА. Одним з алгоритмів, що використовуються в разі *RL*-навчанні, є мережа *Q*, яку можна об'єднати з різними нейронними мережами для визначення оптимального значення дії. Алгоритм спирається на безперервне оновлення станів БПЛА на основі даних, отриманих від бортових датчиків, а потім на визначення оптимальної дії, яку необхідно зробити, і відповідного значення винагороди.

Загалом автономна навігація у невідомому чи невизначеному середовищі є одним із складних завдань для БПЛА. Щоб вирішити цю проблему, необхідно мати методи керування високого рівня, які можуть здійснювати навчання і адаптуватися до умов, що змінюються. Одним із найперспективніших фреймворків для досягнення такої мети є посилене навчання. Описано новий *Model-Based* алгоритм посиленого навчання з моделлю *TEXPLORE*[14], розроблений як метод керування високого рівня для автономної навігації БПЛА [24].

TEXPLORE [25, 26] — ефективний за вибіркою алгоритм з *Model-Based RL* посиленого навчання в реальному часі на основі динамічної моделі середовища, яку можна адаптувати стосовно її функційності, що поєднує інтеграцію планування та навчання. Інтеграція цих двох етапів більш відома як *RL* навчання з моделлю. Нагадаємо, традиційне завдання посиленого навчання [16] пов'язане з навчанням політики керування, що оптимізує числову продуктивність шляхом прийняття рішень поетапно. Поєднання планування та навчання провадиться для оптимізації процесу послідовного прийняття рішень Маркова, *MDP*. Тобто, *TEXPLORE* — це метод навчання агентів виконанню послідовних завдань прийняття рішень за допомоги взаємодії з довкіллям. Підходи до навчання динамічних моделей охоплюють вирішення таких проблем, як робота зі стохастичністю, не-

[14] *TEXPLORE* було випущено публічно як пакет *ROS* за адресою — *http://www.ros.org/wiki/rl-texplore-ros-pkg*

визначеністю, частковістю спостереження та тимчасовою абстракцією [25, 26].

TEXPLORE застосовує факторизовану[15] модель, роблячи окремий прогноз про наступне значення кожної функції стану та винагороди. Він будує дерева рішень для моделювання кожної функції, що дозволяє узагальнювати ефекти дій за станами та динаміку на невидимі стани. Алгоритм *RL* вивчає модель переходу станів та динаміки винагороди, а потім використовує її для планування політики, що дає змогу йому навчатися за меншу кількість дій, ніж у багатьох підходах без моделі. Він також використовує архітектуру реального часу, яка виконує навчання та планування моделі в паралельних потоках, тому агент може діяти в реальному часі. Агент націлюється на дослідження станів, які є багатообіцяльними для остаточної політики, так і невизначеними в моделі. З плануванням на основі вибірок та паралельною архітектурою *TEXPLORE* може безперервно вибирати дії у реальному часі, коли це необхідно. Високої ефективності вибірки він досягає як внаслідок використання властивостей узагальнення дерев рішень при побудові своєї моделі *MDP* та застосування випадкових лісів для обмеження дослідження станами, які є перспективними для швидкого навчання гарної (але не обов'язково оптимальної) політики, замість більш вичерпного аналізу для гарантії оптимальності. Оскільки алгоритм повинен виконувати обмежене дослідження, то дослідження має бути ефективним і націленим на дії станів, які можуть бути перспективними для отримання винагороди.

Ключові ідеї *TEXPLORE*: 1) вивчення кількох доменних моделей, що узагальнюють ефекти дій у різних станах, та цільове дослідження невизначених і перспективних станів; 2) об'єднання пошуку по дереву Монте-Карло та паралельної архітектури для безперервного виконання дій у режимі реального часу.

Як стверджують автори [24], це перший випадок адаптації моделі *TEXPLORE* для автономної навігації квадроторного БПЛА (*Quadrotor UAVs*), який є одним із привабливих типів малих безпілотних літальних апаратів через дуже просту механічну конструкцію та принцип руху. Розроблений алгоритм ретельно протестували з квадрокоптером у середовищі *ROS-Gazebo.* Експериментальні результати показали, що метод здатний вивчити ефективну траєкторію за кілька ітерацій та виконувати дії у реальному часі. Більше того, підхід значно перевершує метод на основі *Q*-навчання.

Відповідно, глибоке посилене навчання, *DRL* призвело до справжнього зсуву в керуванні БПЛА. Ніхто не може заперечувати, що вища точність подання інформації мотивує дослідників докладати додаткових зусиль у використанні цих алгоритмів, які потребують біль-

[15] Факторизація або розкладання на множники — це декомпозиція об'єкта у добуток інших об'єктів.

ше обчислювального часу, ніж класичне керування. Не кажучи вже про те, що *DRL* дає змогу виконувати нові завдання, які раніше були майже неможливими. Той факт, що розроблення нових методів і стратегій *DRL* все ще є постійним процесом з багатьма можливостями, є багатообіцяльною новиною для досліджень у сфері керування БПЛА [1].

В останні роки досягнуто значного прогресу у вирішенні складних проблем у різних галузях за допомогою посиленого глибокого навчання. Відтворення наявних робіт та точна оцінка покращень, що пропонуються новими методами, є життєво важливими для підтримки цього прогресу. На жаль, відтворення результатів для сучасних методів посиленого глибокого навчання не часто буває простим. У роботі [28] надано такі пропозиції щодо того, як забезпечити постійний прогрес у цій галузі, мінімізуючи марні зусилля, які забезпечують через результати, які є невідтворюваними та легко неправильно інтерпретуються.

Висновки

З погляду класифікації підходів до створення мультиагентних систем із застосуванням глибокого посиленого навчання, етапів польотного завдання та основних проблем, з якими стикаються БПЛА: планування шляху, навігація та керування, кожен з цих елементів охоплює ряд підзадач, які вимагають високого рівня керування для бажаного функціонування. Сфера керування БПЛА чи роєм дронів для виконання певних завдань дуже складна. Складність полягає в самій її природі: змішаній динаміці, потребі бути енергетично ефективною та її вразливості до перешкод, шумів датчиків, атмосферних завад, і не модельованої динаміки, взаємодії з невідомими середовищами, що змінюються, тощо. Підходи до навчання динамічних моделей повинні вирішувати такі проблеми як робота зі стохастичністю, невизначеністю, частковістю спостереження тощо. Модель БПЛА важко описати, враховуючи нелінійності в системі. З урахуванням цього найбільш прийнятним для розв'язання проблем керування БПЛА або рою дронів і забезпечення стабільної та плавної навігації БПЛА, запропоновано *DRL* нейронних мереж з підходом без моделей (*Model-Free*) та є декілька робіт зі застосуванням моделі (*Model-Based*).

Останній підхід здійснює оптимізацію процесу послідовного прийняття рішень Маркова за допомоги взаємодії з середовищем і поєднує планування дій агента та його навчання. В цьому разі досягається висока ефективність вибірки як внаслідок вивчення моделі переходу станів та динаміки винагороди, з подальшим її використанням для планування політики, що дозволяє навчатися за меншу кількість дій, ніж у багатьох підходах без моделі та цільового дослідження невизначених та перспективних станів.

На жаль, майже всі роботи мають лабораторний характер, їм бракує перевірки у реальних чи наближених до них умов навколишнього середовища.

Подальші дослідження, на наш погляд, треба зосередити на рішеннях із застосуванням моделі (*Model-Based*) та приділити увагу проєктування типових середовищ, які відповідають тим чи іншим умовам виконання завдання. Такі моделі, можливо, буде простіше і швидше адаптувати до реального оточення.

ЛІТЕРАТУРА / REFERENCES

1. Azar AT, Koubaa A., Ali Mohamed N. et al. Drone Deep Reinforcement Learning: A Review. *Electronics* 2021, Vol. 10(9), 1–30. https://doi.org/10.3390/electronics10090999
2. Oursatyev O.A, Volkov, O.Ye, Tkalya V.H. Automated Machine Learning. State and Prospects Development. *Information Technologies and Systems*, Vol. 2(2), 3–33. [In Ukrainian: Автоматизоване машинне навчання. Стан та перспективи розвитку. *Information Technologies and Systems (Інформаційні технології та системи)* https://doi.org/10.15407//intechsys.2025.02.003
3. Schmidhuber J. Deep Learning in Neural Networks: An Overview. *Neural Networks*, 2015, Vol. 61, 85-117. https://doi.org/10.1016/j.neunet.2014.09.003
4. Sutton, R.S., Barto A.G. *Reinforcement Learning: An Introduction.* MIT Press,Cambridge, Massachusetts, London, England, 2018, 526 p. URL: https://www.google.com/url?sa=t&source=web&rct=j&opi=89978449&url=https://www.andrew.cmu.edu/course/10-703/textbook/BartoSutton.pdf
5. Schmidhuber J. Making the World Differentiable: On Using Self-Supervised Fully Recurrent Neural Networks for Dynamic Reinforcement Learning and Planning in Non-Stationary Environments. *IDSIA*, 1990. URL: https://people.idsia.ch/~juergen/FKI-126-90_(revised)bw_ocr.pdf
6. Schmidhuber J. Reinforcement Learning in Markovian and Non-Markovian Environments. *IDSIA*, 1991. URL: https://sferics.idsia.ch/pub/juergen/nipsnonmarkov.pdf
7. Arulkumaran K. et al. Deep reinforcement learning: A brief survey, 2017. https://doi.org/10.1109/MSP.2017.2743240
8. Zhang H., Yu T. Taxonomy of Reinforcement Learning Algorithms. *Deep Reinforcement Learning,* Springer Singapore, 2020. https://doi.org/10.1007/978-981-15-4095-0_3
9. Schmidhuber J. An on-line algorithm for dynamic reinforcement learning and planning in reactive environments. *IJCNN International Joint Conference on Neural Networks*, 1990, Vol. 2, 253-258. https://doi.org/10.1109/IJCNN.1990.137723
10. Nagabandi A. et al. PDDM: Planing with Deep Dynamics Models for Learning Dexterous Manipulation. *Conference on Robot Learning (CoRL),* 2019. URL: https://sites.google.com/view/pddm
11. Poole D.L., Mackworth A.K. *AI Foundations of Computational Agents.* University of British Columbia, TJ Books Limited, Padstow, Cornwall, 2023, 870 p. https://doi.org/10.1017/9781009258227
12. Francois-Lavet V., Henderson P. et al. *An Introduction to Deep Reinforcement Learning*. 2018. https://doi.org/10.1561/9781680835397
13. Ha D., Schmidhuber J. World Models. Can agents learn inside of their own dreams? *NIPS 2018*, March 27 2018, Oral Presentation. https://doi.org/10.5281/zenodo.1207631
14. Anastasis Germanidis. Introducing General World Models. URL: https://research.runwayml.com/introducing-general-world-models [Accessed Dec. 2023]

15. Gronauer S., Diepold K. Multi-agent deep reinforcement learning: a survey. *Artificial Intelligence Review,* 2021, 1–49. URL: https://link.springer.com/article/10.1007/s10462-021-09996-w
16. Sutton R. S., Barto A. G. *Reinforcement Learning: An Introduction.* A Bradford Book, The MIT Press Cambridge, Massachusetts, London, England,2015, 1–337. URL: https://web.stanford.edu/class/psych209/Readings/SuttonBartoIPRLBook2ndEd.pdf
17. Littman M.L. Markov games as a framework for multi-agent reinforcement learning. *Machine Learning Proceedings,* 11-th International Conference, Rutgers University, New Brunswick, NJ, 10–13 Jul. 1994, 157-163. https://doi.org/10.1016/B978-1-55860-335-6.50027-1
18. Fully Observable vs. Partially Observable Environment in AI. URL: https://www.geeksforgeeks.org/fully-observable-vs-partially-observable-environment-in-ai/ [Accessed May. 2024]
19. Tožička J., Szulyovszky B., de Chambrier G. et all. Application of deep reinforcement learning to UAV fleet control. *SAI Intelligent Systems Conference*, London, UK, 5–6 Sept. 2018, 1169–1177. https://doi.org/10.1007/978-3-030-01057-7_85
20. Liu C.H. et al. Energy-efficient UAV control for effective and fair communication coverage: A deep reinforcement learning approach. *IEEE J. Sel. Areas Commun*., Aug. 2018, Vol. 36 (9), 2059–2070. https://doi.org/10.1109/JSAC.2018.2864373
21. Yang J. et al. Application of reinforcement learning in UAV cluster task scheduling. *Future Gener. Comput. Syst.* 2019, Vol. 95, 140–148. https://doi.org/10.1016/j.future.2018.11.014
22. Wang C. et all. Autonomous navigation of UAV in large-scale unknown complex environment with deep reinforcement learning. I*EEE Global Conference on Signal and Information Processing (GlobalSIP)*, Montreal, QC, Canada, 14–16 Nov. 2017, 858–862. https://doi.org/10.1109/GlobalSIP.2017.8309082
23. Huang Y., Wei G., Wang, Y. V-D D3QN: the Variant of Double Deep Q-Learning Network with Dueling Architecture. *37th Chinese Control Conference (CCC)*, Wuhan, China, 2018, 9130–9135. https://doi.org/10.23919/ChiCC.2018.8483478
24. Imanberdiyev N., Fu C., Kayacan E., Chen I.M. Autonomous navigation of UAV by using real-time model-based reinforcement learning. *14th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, Phuket, Thailand, 13–15 Nov. 2016, 1–6. https://doi.org/10.1109/ICARCV.2016.7838739
25. Hester T., Stone P. TEXPLORE: Real-Time Sample-Efficient Reinforcement Learning for Robots. AAAI Technical Report SS-12-02. *Designing Intelligent Robots: Reintegrating AI.* Department of Computer Science, The University of Texas at Austin. URL: https://cdn.aaai.org/ocs/4271/4271-19461-1-PB.pdf
26. Moerland T.M. *Model-based Reinforcement Learning: A Survey.* 2020, 67 p. URL: http://arxiv.org/pdf/2006.16712
27. Bou-Ammar H., Voos H., Ertel W. Controller Design for Quadrotor UAVs using Reinforcement Learning. *IEEE International Conference on Control Applications*, Yokohama, Japan, 8–10 Sept. 2010, 2130–2135. https://doi.org/10.1109/CCA.2010.5611206
28. Henderson P. et al. Deep Reinforcement Learning that Matters. Thirthy-Second AAAI Conference On Artificial Intelligence (AAAI), 2018. https://doi.org/10.1609/aaai.v32i1.11694

Received 17.06.2025

O.A. OURSATYEV, PhD (Engineering), Senior Researcher,
Institute of Information Technologies and Systems of the NAS of Ukraine,
40, Hlushkova Akad. ave, Kyiv, 03187, Ukraine
https://orcid.org/ 0009-0009-8323-0525
aleksei@irtc.org.ua

O.Ye. VOLKOV, PhD (Engineering), Senior Researcher, Director,
Institute of Information Technologies and Systems of the NAS of Ukraine,
40, Hlushkova Akad. ave, Kyiv, 03187, Ukraine
https://orcid.org/0000-0002-5418-6723
alexvolk@ukr.net

APPROACHES TO CREATING MULTIAGENT SYSTEMS
AND DEEP REINFORCEMENT LEARNING OF DRONES

Introduction. Unmanned aerial vehicles (UAVs) are increasingly used in many complex and diverse tasks related to civil and military spheres. UAVs are a class of aircraft, commonly referred to as drones. They can fly without the presence of a human pilot on board. However, there are a number of unsolved problems with UAVs development: flight path planning, navigation and control. In complex systems, which certainly include UAVs, artificial intelligence (AI) is usually used to solve these problems and ensure the required of its functioning, implemented by the method of deep learning with reinforcement. Modern foreign experience in the use of analytical platforms for controlling mobile objects, in particular UAVs, allows for the use of deep neural networks for the above tasks.

The purpose of the paper is to introduce domain experts whose primary job function is outside of machine learning to the challenges of applying AI to these problems, robust and complex deep neural networks and their training, which remains challenging and requires large amounts of data and practical experience. This can be a form of citizen science and will contribute to the replication of research and the democratization of AI.

Results. An analysis of solutions to these problems using deep reinforcement learning is performed, in particular, control of a swarm of UAVs etc. and a taxonomy of Model-Free deep reinforcement learning algorithms applied in UAV tasks is given. The first experience of solutions using the environment model is considered. Unfortunately, almost all works are of a nature, they lack verification in real or close to them environmental conditions. This paper presents a brief overview of approaches to solving problems of reinforcement learning - interactions between agents and the environment in the process of step-by-step decision making. This approach is applied to solving problems of moving objects and complex and partially observable environments; model-free and model-based learning; mathematical formalization of solving UAV problems under reinforcement learning, including paradigms for learning agents in a multi-agent environment Multi-Agent reinforcement learning. Problems arising in the multi-agent field, such as non-stationarity of the environment from the point of view of a single agent, relative overgeneralization and the problem of assigning credits are discussed. Formal concepts underlying these Multi-Agent reinforcement learning are presented.

Conclusions. An overview of methods for solving the problem of reinforced learning is presented, as a result of which the authors conclude that further research should focus on solutions using a model (Model-Based) and pay attention to the design of typical environments that meet certain conditions for performing the task. Such models may be easier and faster to adapt to the real environment.

Keywords: *unmanned moving objects, UAVs, UAV swarm control, swarm of UAVs, deep reinforcement learning, DRL, world models, world models introduces a model-based approach to RL, training paradigms execution sheme.*