ISSN 2224-087X. Електроніка та інформаційні технології. 2024. Випуск 28. С. 109–122
Electronics and information technologies. 2024. Issue 28. P. 109–122

УДК 004.93 DOI: https://doi.org/10.30970/eli.28.10

ВИКОРИСТАННЯ АЛГОРИТМУ REINFORCEMENT LEARNING ДЛЯ ОПТИМІЗАЦІЇ ПОЛЬОТІВ БПЛА

О. Дуцяк, В. Юзевич

Факультет електроніки та комп'ютерних технологій
Львівський національний університет імені Івана Франка,
вул. Драгоманова, 50, 79005 Львів, Україна
oleg.dystak@gmail.com, yuzevych@ukr.net

Безпілотні літальні апарати (БПЛА) стали переконливим напрямком сучасних досліджень, пропонуючи теми як для цивільних, так і для військових застосувань, таких як перевірка, доставка, розвідка та спостереження. Сфери інтересів включають автономну навігацію, планування шляху польоту в реальному часі та розпізнавання об'єктів. Однак автономний політ пов'язаний з низкою проблем для БПЛА, включаючи розробку стратегій керування, налаштування інформативних параметрів, адаптивне керування, планування траєкторії в реальному часі та розпізнавання об'єктів у невизначених середовищах. В останні роки впровадження машинного навчання набуло популярності як багатообіцяючий підхід до розв'язання відповідних завдань, які мають відношення до автономного польоту БПЛА. Машинне навчання надає розробникам БПЛА здатність оптимізувати розвідувальний вогневий комплекс військ та робити прогнози на основі історичних даних, тим самим вирішуючи потребу в попередньо запрограмованих інструкціях під час автономного польоту.

У даній роботі основна увага фокусується на дослідженні методу машинного навчання, який має назву навчання з підкріпленням (Reinforcement Learning – RL) з оглядом на оптимізацію польоту БПЛА завдяки оптимізації (удосконаленні) процесу реалізації його шляху. Було запропоновано метод планування траєкторії БПЛА на основі градієнтного алгоритму RL. У відповідному алгоритмі представлено принципи та компоненти методу навчання з підкріпленням. Завдання раціональному плануванні траєкторія БПЛА реалізується за допомогою глибокого навчання з підкріпленням, а допоміжне тривимірне середовище описується за допомогою растрових методів для побудови растрової карти. Дослідження результатів функціональності запропонованого алгоритму було проведено в середовищі тривимірного моделювання. Алгоритм RL використовується для визначення оптимальної політики і при цьому для кожного растру розглядаємо специфічний стан, 26 можливих напрямків як набори дій, а функцію значення дії як цільову функцію.

Під час вивчення результатів проведеного експерименту було виявлено, що після 160 експериментів, завдяки використанню градієнтного методу RL вдалося визначити шлях від початкового положення до кінцевої точки та успішно подолати усі перешкоди з кроком у часі 84 секунди. Перспективи подальшого удосконалення запропонованого методу можуть включати дослідження методів, щоб забезпечити більш гнучке переміщення БПЛА в межах карти та провести оптимізацію ефективності вибірки для покращеної застосовності RL в реальному світі.

Ключові слова: автономні системи, машинне навчання, польотна навігація, планування шляху, навчання з підкріпленням, алгоритми керування.

Вступ.

Використання технологій штучного інтелекту в сфері безпілотних літальних апаратів (БПЛА) забезпечує підвищення безпеки, адаптивності, оптимізації, ефективності, потужності. Однак застосування пов'язаних з цим технік потребує нових навичок і можливостей, а також адекватної адаптації відповідних правил. Потрібно нові концепції стандартизації, сертифікації, кваліфікації та обміну даними. Однією з найважливіших підгалузей штучного інтелекту є машинне навчання, яке зосереджене на розробці алгоритмів і моделей, які дозволяють комп'ютерним моделям навчатися на основі наданих наборів даних і робити прогнози чи приймати рішення без явного програмування. За своєю суттю, машинне навчання спрямоване на створення систем, які можуть автоматично навчатися та вдосконалюватися на основі вже набутого досвіду. Іншими словами, у процесі тренування машинне навчання генерує знання із наявних даних про закономірності, зв'язки, залежності та приховані структури цих даних. Відповідна концепція має широкий потенційний спектр застосувань у сфері безпілотної авіації. Завдяки своїй можливості працювати з великими даними та їх складністю, а також швидкій і високій точності обробки, машинне навчання використовується для вдосконалення процесів навігації та польоту БПЛА.

Задля більш загального розуміння, необхідно зазначити, що концепція машинного навчання ділиться на три основних підгалузі: контрольоване навчання; навчання без нагляду; навчання з підкріпленням. При контрольованому навчанні, модель навчається на міченому наборі даних, який має як вхідні, так і вихідні параметри. У контрольованому навчанні алгоритми допомагають у процесі навчання відображати точки між входами та правильними виходами. Провідний механізм навчання без нагляду, навпаки, полягає у виявленні з допомогою алгоритму закономірностей та зв'язків, використовуючи немічені дані. На відміну від контрольованого навчання, даний тип навчання не передбачає надання з допомогою алгоритму позначених цільових результатів. Основною метою навчання без нагляду часто є виявлення прихованих закономірностей, подібностей або кластерів у даних. Останній тип машинного навчання є навчання з підкріпленням, що являє собою метод, який взаємодіє з навколишнім середовищем, виконуючи певні дії та виявляючи помилки. Проби, помилки та затримка являють собою найбільш важливі характеристики навчання з підкріпленням. Кожного разу, коли проходить відбір даних, модель опрацьовує і додає дані до своїх знань, що є навчальними даними. Отже, чим більше часу навчається модель, тим кращу вона отримує підготовку і стає більш досконалою.

Робота [1] була зосереджена на проведенні дослідження, яке було спрямоване на впровадження прогнозування траєкторії в реальному часі шляхом інтеграції методів машинного навчання в систему спостереження UTM за допомогою Broadcast Remote ID. Задля полегшення даної операції, було розроблено саморобний приймач для трансляції Remote ID з використанням мікроконтролера ESP32. Згодом було проведено два різних льотних випробування з використанням саморобного навчального БПЛА. У першому польоті дані про траєкторію БПЛА використовуються як навчальні дані для алгоритмів машинного навчання. Другий політ був зосереджений на реалізації прогнозування траєкторії в реальному часі. Модель прогнозування траєкторії використовує такі вхідні дані, як широта, довгота, висота, швидкість, напрямок, час затримки та індикатор потужності отриманого сигналу (Received Signal Strength Indicator – RSSI). Завдяки аналізу офлайн-даних першого польоту встановлено, що алгоритм Gated Recurrent Unit

(GRU) має перевагу перед алгоритмом Long Short-Term Memory (LSTM) [1]. Результати другого льотного випробування довели можливість прогнозування траєкторії GRU. Модель GRU успішно створює прогнози в режимі реального часу під час польоту БПЛА, демонструючи рівні точності, подібні до тих, що були отримані в автономному режимі аналізу прогнозів за короткий час обробки [1]. Було зроблено висновок відносно того, що відповідне прогнозування траєкторії в реальному часі могло б підвищити безпеку роботи БПЛА, з урахуванням розрахункової позиції, коли час затримки перевищує порогове значення.

В ході проведеного дослідження у роботі [2] було виявлено, що використання інфраструктури 5G, щоб дозволити БПЛА орієнтуватися в складних середовищах незалежно від GPS та інших сигналів, що передаються БПЛА, має потенціал до оптимізації польоту при умові використання методів машинного навчання для розв'язання сформульованої проблеми планування траєкторії. Зокрема, були запропоновані дві методики, що ґрунтуються на машинному навчанні, а саме, підходи на основі навчання з підкріпленням і глибокого контрольованого навчання. Потім було проведено аналіз ефективності кожного із запропонованих методів у порівнянні з підходами на основі оптимізації. Результати моделювання показують, що запропоновані рішення на основі підкріплення та глибокого контрольованого навчання забезпечують майже оптимальні рішення для сформульованої проблеми планування траєкторії з порівнянною точністю 99% і 98%, відповідно, порівняно з оптимальною межею під час дотримання в режимі реального часу вимоги розрахунку [2].

Мета даної роботи пов'язана з удосконаленням алгоритму RL для розробки сучасних методів оптимізації польотів БПЛА, фокусуючись на використанні концепції штучного інтелекту, зокрема, з урахуванням підходів машинного навчання.

Матеріали та методи.

Загалом, оптимізація польоту БПЛА з можливим доцільним використанням технологій машинного навчання передбачає вдосконалення кількох ключових атрибутів для підвищення продуктивності, ефективності, безпеки та надійності літального апарату. До найважливіших основних атрибутів відносять:

- автономність, підвищення якої дозволяє апаратам виконувати більше завдань без втручання людини, таких як автономний зліт і посадка, навігація за маршрутними точками, уникнення перешкод і планування місії;
- системи візуалізації, тобто радари та інфрачервоні сенсори, покращення роботи яких позитивно впливає на ситуаційну обізнаність, виявлення перешкод і можливості зондування з допомогою БПЛА навколишнього середовища.
- удосконалення алгоритмів систем зв'язку забезпечує покращену потокову передачу даних у реальному часі, дистанційне керування та функціональні можливості командування та контролю.
- планування траєкторії та навігація, тобто алгоритми оптимізації маршруту, інструменти планування місій та можливості обробки геопросторових даних.

У відношенні оптимізації польотів БПЛА навчання під наглядом часто передбачає використання навчального набору даних $X \cup Y$, у якому кожному запису i відповідають ознаки $x_i \in X$ (вхідні змінні) і пов'язані з ними результати спостережень $y_i \in Y$ (вихідні змінні, також звані як «мітки»). На основі відповідних даних використовується модель типу h_θ, щоб спробувати наблизити функцію, яка відображає множину X на Y якомога

точніше: $h_\theta: X \to Y$. Концепція «контрольованого навчання» походить від ітераційного процесу, у якому модель генерує прогнози для будь-якого вхідного сигналу, порівнює ці прогнози з очікуваними результатами та відповідно уточнює свої параметри (наприклад, за допомогою оптимізації процедури градієнтного спуску).

Стратегії навчання під наглядом є універсальними та знаходять застосування як у задачах регресії, так і в класифікації в області оптимізації польоту БПЛА. У завданнях регресії метою є наближення неперервної функції, яка ефективно фіксує зв'язок між вхідними характеристиками та спостережуваною вихідною змінною. Наприклад, до таких завдань можна віднести прогнозування тривалості польоту або часу автономної роботи (спостереження) на основі сезонних коливань, географічних факторів або демографічних даних (особливостей). З іншого боку, у сценаріях класифікації метою є апроксимація функції, яка категоризує вхідні дані і допомагає формувати окремі класи. До відповідних результатів можна віднести двійкові класифікації, такі які допомагають визначити умови того, чи погодні умови є провідними (рівні 1) чи несприятливими (рівні 0) для роботи БПЛА, а також небінарні класифікації. Прикладом останнього варіанту є навчання системи розпізнавання зображень для ідентифікації об'єктів на аерофотознімках на основі піксельних даних (об'єктів) і попередньо визначеного набору класів.

Методи неконтрольованого навчання, на відміну від навчання під наглядом, яке спирається на мічені набори даних із відомими результатами, працює виключно на вхідних функціях (X), що робить їх особливо актуальним у сценаріях, коли мічені дані можуть бути дефіцитними або недоступними [2]. Одним із доречних застосувань неконтрольованого навчання для оптимізації польоту БПЛА може бути виявлення магнітних аномалій.

Використовуючи процедуру вимірювання відстані з урахуванням подібності, алгоритми неконтрольованого навчання можуть ідентифікувати ненормальні шаблони або викиди в даних польоту, такі як несподівані відхилення від типових траєкторій польоту або незвичні показання сенсорів. Використання кластерного аналізу є одним з актуальних застосувань для оптимізації польоту БПЛА. Оцінюючи схожість і відмінності в даних польоту, алгоритми неконтрольованого навчання можуть групувати подібні польоти разом на основі різних критеріїв, таких які пов'язані з цілями місії, умовами навколишнього середовища або траєкторії польоту. Відповідні кластери допомагають надавати цінну інформацію про загальні схеми польотів і профілі місій, дозволяючи операторам оптимізувати маршрути, ефективніше розподіляти ресурси та покращувати загальне планування та виконання місій. Крім того, методи зменшення розмірності відіграють важливу роль в оптимізації польоту БПЛА завдяки спрощенню складних наборів даних із збереженням важливої інформації. Стискаючи великі багатовимірні вхідні дані в менший вимірний простір, методи зменшення розмірності, як-от аналіз головних компонентів (PCA), допомагають визначити ключові змінні БПЛА або характеристики, які мають найбільш значний вплив на характеристики польоту.

У навчанні з підкріпленням модель, яку потрібно навчити, називається «агентом», дані, що надходять у модель, називаються «станом», а вихідні дані моделі називаються «дією» [2]. Крім того, процес RL та його моделі, підметоди та структури даних також підпадають під термін агент. Основна ідея RL полягає в тому, щоб навчити агента взаємодіяти з його середовищем. Агент представлений моделлю машинного навчання, наприклад таблицею, логікою на основі правил, параметризованою функцією або штучною нейронною мережею. Агент перетворює стани, які є вхідними змінними, у дії, що є вихідними. Кожного разу, коли агент виконує дію, середовище переходить у

наступний стан, для якого агент повинен вибрати наступну дію. На відміну від навчання під наглядом, для застосування методів RL не потрібні очікувані результати. Функція винагороди оцінює дії агента і присвоює їм позитивні або негативні значення. Мета агента пов'язана з максимізацією отриманих винагород або мінімізацію отриманих втрат з часом. Таким чином агент вчиться передбачати найкращу дію для заданої умови.

У контексті оптимізації польоту БПЛА на основі методології RL автономним БПЛА дозволено вивчати та адаптувати свою поведінку в режимі реального часу на основі факторів навколишнього середовища. Використовуючи підкріплене навчання, БПЛА можуть динамічно коригувати параметри польоту, такі як висота, швидкість і вибір маршруту, щоб оптимізувати цілі місії, максимізуючи ефективність і безпеку експлуатації. Як правило, БПЛА оснащені камерами, а також сенсорами, які збирають інформацію з навколишнього середовища, що допомагає БПЛА самостійно переміщатися в заданому середовищі. Навігація БПЛА зазвичай проводиться у віртуальному 3D-середовищі, оскільки БПЛА мають обмежені обчислювальні ресурси та джерело живлення. RL може допомогти розв'язувати різноманітні проблеми, коли агент виступає як людина-експерт у домені. Агент взаємодіє з середовищем, відповідаючи дією та отримуючи винагороду. Камери та сенсори БПЛА у процесі польоту збирають інформацію з навколишнього середовища для адекватного представлення стану. Функцію винагороди в RL для БПЛА можна адаптувати для визначення пріоритетів таких факторів, як споживання енергії, час виконання місії та уникнення перешкод, дозволяючи БПЛА автономно керувати складними середовищами та адаптуватися до мінливих зовнішніх умов. Завдяки неперервній взаємодії з оточенням і вивченню минулого досвіду БПЛА, оснащені алгоритмом RL, можуть покращити свої можливості прийняття рішень і досягти оптимальної продуктивності в різних сценаріях польоту.

В межах поточного дослідження, задля оптимізації польоту БПЛА розглядатимуться різні підметоди навчання з підкріпленням. Тип RL машинного навчання поділяється на градієнт-залежне, градієнт-незалежне та гібридне навчання. Градієнт-залежні методи RL навчають агента з урахуванням градієнтів його параметрів, які можна обчислити на основі різниці між виходом агента та очікуваним виходом, що отримуємо в результаті функції винагороди. Градієнтно-залежні методи RL поділяються на модельні та безмодельні підходи [2].

Градієнтно-залежні підходи RL переслідують мету максимізації суми винагород, отриманих у майбутньому, завдяки вибору відповідних дій, починаючи з поточного моменту часу t. Тут майбутні результати $G_t = R_{t+1} + r_{t+2} + \cdots + R_T$ зазвичай дисконтуються на коефіцієнт $0 < \gamma \leq 1$, який експоненційно змінюється на заданому етапі часу, і таким чином можемо записати [2, 3]:

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots = \sum_{k=1}^{\infty} \gamma^{k-1} R_{t+k} \quad (1)$$

Тут γ – параметр дисконтування (discount factor); R_{t+i} – винагорода, отримана після i-того переходу; R_{t+k} – сумарна отримана винагорода.

З одного боку, дисконтування має математичне та економічне підґрунтя, оскільки в незавершених середовищах, у яких не існує визначеного кінцевого стану, сума отриманих винагород наближалася б до нескінченності без дисконтування. Однак для реалізації методу RL необхідно, щоб сума отриманих винагород наближалася до обмеження. З іншого боку, навіть у часових умовах винагороди часто знижуються, щоб віддати

перевагу винагородам у найближчому майбутньому над винагородами у віддаленому майбутньому.

Оптимальна кумулятивна винагорода G_t^*, яка може бути оцінена відповідно до заданого кроку в часі t, є результатом розгляду послідовних дій, які максимізують загальну отриману винагороду з допомогою співвідношення [4, 5].

$$G_t^* = \max_{\{a_k\}_{k=1}^{\infty}} \sum_{k=1}^{\infty} \gamma^{k-1} R(S_k, a_k). \tag{2}$$

Тут a_k – параметр, який характеризує дію a на кроці k; S_k – інтегральна характеристика середовища в момент часу k; $R(S_k, a_k)$ – функція винагороди (функція, що обчислює винагороду одержану при переході із стану S_k при виборі дії a_k).

Слід зазначити, що дія на кроці часу k, яка забезпечує максимальну винагороду за $k+1$, не обов'язково є оптимальним вибором через те, що різні дії призводять до різних наступних станів. Наприклад, вибір гіршої дії може призвести до наступного стану, в якому найкраща дія обіцяє значно вищу винагороду, ніж максимальна доступна винагорода в інших станах. Це призводить до складного дерева рішень, яке, у крайньому випадку, містить потенційні рішення для часового горизонту, припускаючи, що є стабільні дії для вибору результату в кожному стані, таким чином, простір рішень зростає експоненційно залежно від розглянутого часового горизонту.

Простір станів і дій, а також часовий горизонт зазвичай виявляються занадто великими для того, щоб спробувати всі можливі послідовності рішень, щоб визначити оптимум за розумний проміжок часу. Оптимальна політика прийняття рішень має таку властивість, що, незалежно від початкового стану та першого рішення, решта рішень повинні формувати оптимальну політику залежно від першого рішення отриманого наступного стану. Звідси випливає, що оптимальна політика прийняття рішень складається з оптимальних підполітик. Таким чином, перше рішення відокремлюється в момент часу t від усіх майбутніх рішень, починаючи з $k = (t+1, t+2, \ldots, T)$ [5, 6]:

$$G_t^* = \max_{a_t} R(S_t, a_t) + \gamma \left(\max_{\{a_k\}_{k=1}^{\infty}} \sum_{k=1}^{\infty} \gamma^{k-1} R(S_k, a_k) \right) \tag{3}$$

Формула (2) відповідає правому члену у формулі (3) для $k = t+1$. Формулу (2) для $k = t+1$ у свою чергу можна перетворити на формулу (3). Звідси випливає, що формула (3) є рекурсивною функцією, яку можна спростити наступним чином:

$$v^*(S_t) = \max_{a_t} R(S_t, a_t) + \gamma v^*(S_t + 1) \tag{4}$$

Тут $v^*(S_t)$ для спостережуваного стану S_t представляє загальну винагороду в майбутньому під час дотриманні політики оптимального прийняття рішень. Функція $v^*(S_t + 1)$ називається функцією корисності стану (за оптимальної політики прийняття рішень $*$), оскільки вона оцінює корисність V (Value) стану. Дане рівняння є функціональним рівнянням, тобто рівнянням, що складається з кінцевої кількості невідомих функцій і незалежних змінних. При його розв'язанні визначається невідома функція V, яка, починаючи з початкового стану, дозволяє обчислювати максимально можливу винагороду в майбутньому.

У цьому відношенні проводиться спроба наблизити функцію корисності через взаємодію з навколишнім середовищем і на основі отриманих винагород. Існує два варіанти функцій корисності стану $v_\pi(s)$для оцінки кумулятивної майбутньої винагороди, починаючи від стану s і використовуючи політику прийняття рішень π [8, 9]:

$$v_\pi(s) = E_\pi[G_t|S_t = s] = E_\pi\left[\sum_{k=1}^{\infty} \gamma^{k-1} R_{t+k}|S_t = s\right] \quad (5)$$

$E_\pi[\cdot]$ визначає очікуване значення кумулятивної майбутньої винагороди з часу t, припускаючи, що агент застосовує політику прийняття рішення π. (Далі буде, що є двійкове значення E_{t+1}, яке сигналізує, чи завершується середовище в S_{t+1}).

Другим варіантом є функція корисності дії $q_\pi(s, a)$, яка, починаючи зі стану s, оцінює кумулятивну майбутню винагороду за вибір дії a, за припущення, що політика рішення π буде виконуватися після застосування дії a [5, 6]:

$$q_\pi(s, a) = E_\pi[G_t|S_t = s, A_t = a] = E_\pi\left[\sum_{k=1}^{\infty} \gamma^{k-1} R_{t+k}|S_t = s\,, A_t = a\right] \quad (6)$$

Наступний крок полягає у проведені процедур оцінки корисності дій, метою яких є вивчення функції корисності дії $q_\pi(s, a)$ для задачі послідовного прийняття рішень. Фактична політика прийняття рішень виводиться з функції корисності вивченої дії. Дія вибирається з найбільшою прогнозованою вигодою, тобто з найвищою прогнозованою сукупною винагородою в майбутньому. Припускається, що простір дій агента є кінцевим і дискретним. Для заданої умови оцінюється корисність кожної можливої дії. Агент вибирає або дію з найвищою корисністю, або випадковий вибір дії з найвищою корисністю і представляє політику рішення, отриману від функції корисності дії. Вибір випадкових дій необхідний для дослідження простору рішень. Проте з плином часу ймовірність вибору випадкової дії зменшується.

У процесі навчання взаємодія між агентом і його середовищем породжує спостереження, які акумулюються в пам'яті досвіду. Спостереження являють собою кортеж розмірністю 5, що складається зі стану S_t, вибраної дії A_t, отриманої винагороди R_{t+1}, результуючого подальшого стану S_{t+1} і двійкового значення E_{t+1}, яке сигналізує, чи завершується середовище в S_{t+1}. Кожного разу, коли до пам'яті досвіду додається нове спостереження (і якщо пам'ять досвіду вже містить достатню кількість спостережень), виконується черговий етап навчання. Для цього пакет із випадковими спостереженнями генерується з пам'яті досвіду S_t, а наступні стани S_{t+1} поширюються вперед через дві різні параметризовані функції корисності $q_\pi(s, a; \theta)$ і $q_\pi(s, a; \theta^-)$, які з допомогою ШНМ використовуються для визначення відповідних утиліт дії Q_t і Q_{t+1}. В даному випадку θ та θ^- – вектори, які є набором параметрів моделі та визначають її поведінку.

Обидві ШНМ спрямовані на апроксимацію функції корисності дії q_π. ШНМ $q_\pi(s, a; \theta)$ представляє агента, який підлягає навчанню, який також слід використовувати поза процесом навчання. $q_\pi(s, a; \theta^-)$ є копією $q_\pi(s, a; \theta)$, чиї параметри містять максимум кроків часу c в минулому $q_\pi(s, a; \theta)$. ШНМ $q_\pi(s, a; \theta^-)$використовується лише під час навчання. Q_t і Q_{t+1} є $|A|$-вимірними векторами, де кожен елемент вектора описує очікувану кумулятивну винагороду в майбутньому за вибір відповідної дії. Q_t і Q_{t+1} використовуються для обчислення сигналу помилки, який повертається ШНМ $q_\pi(s, a; \theta)$

за допомогою зворотного поширення помилки, яке являє собою алгоритм, що складається з фази ініціалізації та трьох повторюваних фаз. По-перше, усі ваги та зміщення ініціалізуються випадковими значеннями. Наступні три кроки повторюються, доки не буде перевищено попередньо визначене значення помилки або досягнуто максимальної кількості повторень. Максимальне значення похибки має бути встановлено таким чином, щоб створювалося співвідношення високої можливості узагальнення та малого перетренування мережі. Три повторювані фази можна описати таким чином:

- з набору навчальних даних вибирається пара елементів. Вхідні дані (особливість) подаються в мережу, яка створює з них вихідні дані;
- цей результат порівнюється з бажаним результатом (мітка). З нього обчислюється значення помилки;
- значення помилки використовується для коригування ваг і зміщення, яке потім поширюється по мережі.

На цьому фоні для обчислення Q_t і Q_{t+1} використовуються різні параметризовані ШНМ, щоб уникнути небажаних кореляцій між Q_t і Q_{t+1} і пов'язаних з ними коливань і розбіжностей у процесі навчання.

Сигнал помилки обчислюється в два етапи. На першому із них цільове значення U_t для вибраної дії Q_t, A_t оцінюється таким чином:

$$U_t = R_{t+1} + (1 - E_{t+1})\gamma \, max \, Q_{t+1} \tag{7}$$

Якщо середовище завершується на S_{t+1} , при чому $E_{t+1} = 1$, тоді лише винагорода R_{t+1} використовується як навчальна мітка. В іншому випадку також додається знижена максимальна очікувана винагорода для наступного стану S_{t+1}. Похибка вихідного шару L агента розраховується наступним чином:

$$\delta^{(L)} = (U_t - Q_t, A_t)^2 \tag{8}$$

Функції $q_\pi(s, a; \theta)$ визначаються за допомогою методу зворотного поширення помилки, а параметри ШНМ оновлюються. Описаний процес навчання характеризується тим фактом, що при обчисленні сигналу помилки враховується не фактична кумулятивна винагорода в майбутньому G_t, а лише прогнозована вигода від дії на наступному кроці часу $t + 1$. Процес навчання повторюється для кожного кроку протягом кількох епізодів. Апроксиматор функції корисності цільової дії $q_\pi(s, a; \theta^-)$ оновлюється шляхом заміни його параметрів після n часових кроків (де n вільно вибирається, але має бути більше одиниці) на параметри апроксиматора функції корисності дії $q_\pi(s, a; \theta)$. Алгоритм навчання агента зображено на рис. 1.

Основна мета розробки алгоритму навчання з підкріпленням (типу RL) для автономної навігації БПЛА полягає в тому, щоб створити модель, яка здатна здійснювати навігацію від початкової точки A до цільової точки B, вибираючи невизначений шлях P(A, B) і без зіткнення з будь-якою перешкодою під час польоту. У цьому випадку припускають, що можливий шлях P розкладається на серію етапів $p_1, \ldots, p_N$, де кожен p_i , $i \in 1..N$ відноситься до маневру уникнення перешкоди. Перешкодами на маршруті можуть бути будівлі або будь-які інші об'єкти, які з'являються на шляху БПЛА. Для цього він має завчасно виявляти перешкоди, сприймати їх місце розташування, позицію та можливий рух і уникати зіткнень.

Важливо сформулювати навігаційне завдання БПЛА і визначити основні складові рішення RL, тобто простір спостережень, простір дій та функцію винагороди. Простір спостережень відповідає формальному представленню світу і БПЛА в ньому. Стан БПЛА являє собою знімок усіх цих параметрів у певний момент часу. Стан повинен містити якомога більше інформації про можливості виконання завдання.

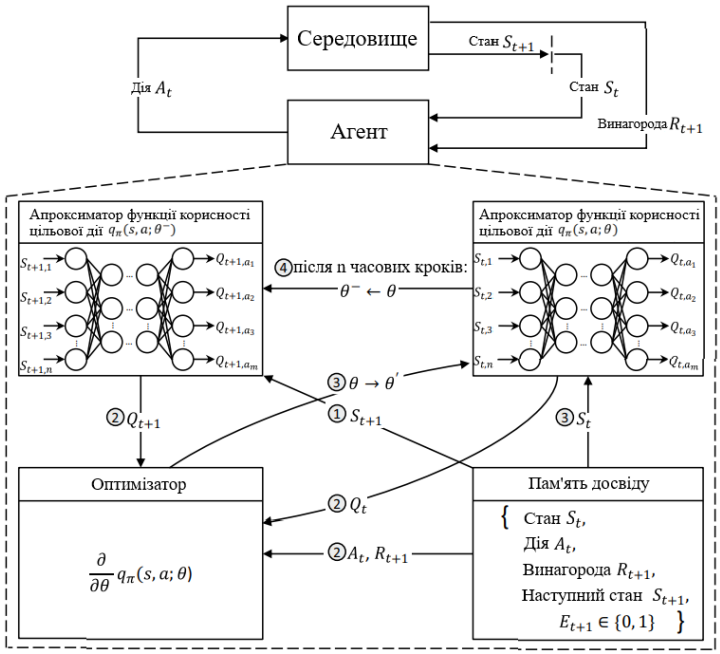

Рис. 1 – Схема алгоритму навчання агента
Fig.1 – Scheme of the agent training algorithm

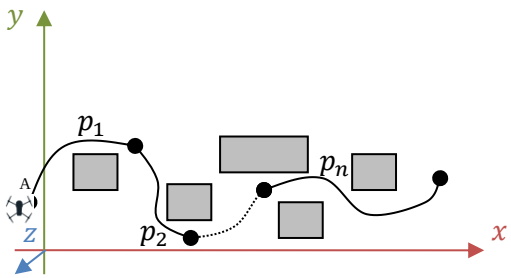

Рис. 2 – Схема маршруту БПЛА: вид зверху
Fig. 2 – UAV flying path sample: top view

Дотримуючись прикладу, зображеного схематично на рис. 2, видно, що БПЛА може рухатися по будь-якій із трьох осей, тобто у напрямках x, y та z. У цьому випадку як параметри стану використовуються:

- відносне положення від поточного положення A до точки призначення B, яке позначається трійкою відстаней на кожній осі ($D_{(A,B)x}$, $D_{(A,B)y}$, $D_{(A,B)z}$);
- швидкості БПЛА за трьома осями (V_x, V_y, V_z);
- орієнтація БПЛА за трьома осями (P, Y , R), відповідно для тангажу, повороту та крену;
- поточна висота;
- вимірювання відстані з двох сенсорів, які прикріплені спереду та знизу дрона (D_{front}, D_{bottom}).

Висоту використовують для того, щоб навчити агента вибирати прийнятні значення: висоти для польотів; швидкості - щоби навчити агента наближатися до перешкод і цілей з відповідною швидкістю та уникати зіткнень або перельотів. Орієнтація використовується для того, щоб знати, куди орієнтований літальний апарат у будь-який момент. У поєднанні з переднім сенсором відстані він поступово дозволить агенту дізнаватися, де знаходяться перешкоди та як їх обережно уникати.

Простір дій містить усі доступні дії, які агент може виконати в будь-який момент. Враховуючи, що БПЛА може рухатися в усіх трьох вимірах, визначають шість різних дій - по дві на вісь. Зокрема, було вирішено збільшити швидкість БПЛА на значення за замовчуванням в осях: -Z і $+Z$ для контролю висоти, $-X$ і $+X$ для руху вперед або назад і, нарешті, $-Y$ і $+Y$ для руху ліворуч і праворуч. Враховуючи простори спостереження та дій, метою агента є вибір найбільш правильної дії в кожен момент шляхом оцінки стану БПЛА. Процес навчання оцінює кожну дію після її виконання та обчислює винагороду, яку має бути надано агенту за допомогою функції винагороди.

У випадку з БПЛА функція винагороди змушує дрон мінімізувати відстань між поточною та цільовою позицією без зіткнення з будь-яким об’єктом або землею. Отже, винагорода за дію, яка не призвела до зіткнення, є функцією відстані до цілі, отриманої в результаті дії. Негативна винагорода дається у випадку зіткнення, що відповідає відстані дрона від цілі, коли зіткнення сталося. Коли дрон досягає цільової зони, видається максимальна позитивна винагорода. Ще дві негативні нагороди даються, коли БПЛА летить, не досягаючи цілі, і коли він виходить за межі дислокації цільового об’єкту в будь-якому напрямку.

Результати й обговорення.

У процесі планування шляху алгоритмом RL вивчається та аналізується модель середовища. Дослідження в даній роботі проводяться в середовищі тривимірного моделювання, для того, щоб забезпечити точність планування та в той же час забезпечити безпеку БПЛА.

Запишемо критеріальні співвідношення для залежностей від відповідних параметрів функції корисності $q_\pi(s,a;\theta)$, ваги конструкції $P(b,c,h)$, а також міцності матеріалу конструкції (structural strength) $S(*)$[14, 15];

$$q_\pi(s,a;\theta) \Rightarrow \max; \quad P(b,c,h) \Rightarrow \min;$$

$$S(*) = S(W_S, \sigma_S, \gamma_{ad}, A_{ad}, W_{PL}, K_{1C}) \Rightarrow \max. \qquad (9)$$

Тут W_S, σ_S – поверхневі енергія та натяг матеріалу конструкції (дрона) відповідно; γ_{ad} – енергія розриву адгезійних зв'язків; A_{ad} – робота адгезії; b, c, h – геометричні розміри конструкції (довжина (b), ширина (c), висота (h)); W_{PL} – енергія поверхневої пластичної деформації (the surface plastic strain energy) матеріалу; K_{1C} – в'язкість руйнування матеріалу (the fracture toughness of the material) [14].

Залежності $q_\pi(s,a;\theta)$, $P(b,c,h)$, а також $S(*)$ приводимо до безрозмірного вигляду і з їх допомогою формулюємо функцію компромісу $F_Q(D,R)$ (Trade-off function) і з її допомогою оптимізаційний критерій, оскільки перші два з них стосовно безпеки CPS повинні мати тенденцію до зменшення, а третій та четвертий за абсолютною величиною – до збільшення:

$$F(q_\pi, P, S, R) = \alpha \cdot q_\pi(s,a,\theta,R) + \beta \cdot P(b,c,h,R) + \delta \cdot S(*,R) \Rightarrow opt, \qquad (10)$$

де R – ризики; $q_\pi(s,a;\theta,R)$, $P(b,c,h,R)$, $S(*,R)$ – функціональні залежності, які встановлюємо на основі експерименту, а також обчислювального експерименту; α, β, δ – коефіцієнти вагомості, які визначаємо на основі обчислювального експерименту. В першому наближенні залежності параметрів (10) від ризику R вважаємо лінійними.

Для конструктивних елементів дрона вибирають вуглецеве волокно, для якого поверхнева енергія становить W_S = 41,8 мJ/м^2, дисперсна компонента складає W_{SD} = 21,5 мJ/м^2, а полярна – W_{SP} = 20,3 мJ/м^2; поверхнева енергія отвердженої фенол-формальдегідної матриці – 49,2 мJ/м^2 (зміцненого матеріалу), а дисперсна та полярна компоненти – 22,5 та 26,4 мJ/м^2 відповідно [16].

Вуглецеві волокна мають високу міцність на розтяг (від 3 до 7 ГПа) і високий модуль Юнга (від 200 до 500 ГПа). Модуль пружності вуглецевих волокон зазвичай становить приблизно 228 ГПа (середнє значення), а їх кінцева міцність на розтяг зазвичай складає близько 3.5 ГПа (середнє значення). Вуглецеве волокно демонструє велике середнє значення осьового коефіцієнта Пуассона і приймає значення в діапазоні 0,26-0,28.

Кількість нейронів на вхідному рівні нейронних мереж відповідає розміру стану кожного часового інтервалу, а кількість нейронів на вихідному рівні мережі відповідає розміру простору дії. Як видно на рис. 3, після 160 експериментів, завдяки використанню градієнтного методу RL вдалося визначити шлях від початкового положення до кінцевої точки та успішно подолати усі перешкоди з кроком у часі 84 секунди.

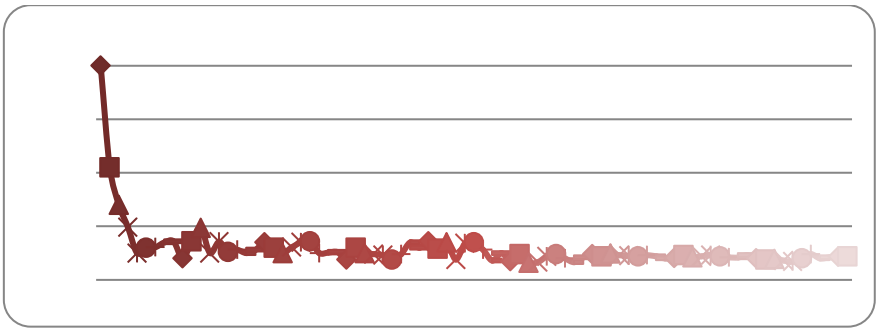

Рис. 3 – Результати навчання на основі градієнтного алгоритму RL
Fig. 3 – Gradient RL algorithm learning results

Висновки.

Загалом, проводячи підсумки проведеного дослідження, у даній роботі було зроблено висновок, що на відміну від звичайних методів, інтеграція теорії RL у планування шляху БПЛА дозволяє освоїти здатність до навчання, подібну до людини. Це виявляється особливо цінним у розв'язанні завдань, що характеризуються високою складністю, невідомим і складним середовищем та невизначеними факторами, що дозволяє БПЛА ефективно орієнтуватися в несподіваних ситуаціях і на складній місцевості для виконання поставлених задач.

Навчання з підкріпленням у поєднанні зі штучними нейронними мережами дозволяє розв'язати задачі, властиві традиційним алгоритмам, такі, як обробка просторів станів і дій великої розмірності, вимоги до ємності зберігання та складності навчання. Будучи технологією самонавчання, вона ефективно підходить для розв'язання задач, з якими стикаються традиційні алгоритми планування траєкторії польоту.

З допомогою функції компромісу сформульовано оптимізаційне співвідношення, в якому враховано функція корисності, вага конструкції і вираз для міцності матеріалу конструкції дрона.

Також у даній статті пропонується метод планування траєкторії БПЛА на основі градієнтного алгоритму RL. Завдання планування шляху реалізується за допомогою глибокого навчання з підкріпленням, а 3D-середовище описується за допомогою растрових методів для побудови растрової карти. Алгоритм RL використовується для визначення оптимальної політики, розглядаючи кожен растр як стан, 26 можливих напрямків як набори дій, а функцію значення дії як цільову функцію.

Результати моделювання демонструють, що алгоритм успішно долає 15 тестових перешкод, згенерованих випадковим чином, та виконує завдання планування шляху приблизно за 84 секунди після 160 ітерацій навчання мережі. Однак, незважаючи на успіх методу в цьому конкретному прикладі, залишаються деякі проблеми та обмеження під час застосування алгоритму RL до реальних систем. Дискретний характер карт, які використано в цій статті, обмежує простір стану та дій кінцевими та дискретними значеннями, накладаючи обмеження на досяжні найкоротші шляхи.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

[1] *Ruseno, Neno & Lin, Chung-Yan*. (2024). Real-Time UAV Trajectory Prediction for UTM Surveillance Using Machine Learning. Unmanned Systems. 10.1142/S230138502550030X

[2] *Afifi, Ghada & Gadallah, Yasser*. (2022). Cellular Network-Supported Machine Learning Techniques for Autonomous UAV Trajectory Planning. IEEE Access. PP. 1-1. 10.1109/ACCESS.2022.3229171.

[3] *Ma, Chao & Qu, Zhi & Li, Xianbin & Zongmin, Liu & Zhou, Chao*. (2023). Machine Learning and UAV Path Following Identification Algorithm Based on Navigation Spoofing. Measurement Science and Technology. 34. 10.1088/1361-6501/acf3da

[4] *Khan, Anamta & Campos, Joao & Ivaki, Naghmeh & Madeira, Henrique*. (2023). A Machine Learning driven Fault Tolerance Mechanism for UAVs' Flight Controller. 10.1109/PRDC59308.2023.00034.

[5] *Fagundes-Junior, Leonardo & de Carvalho, Kevin & Ferreira, Ricardo & Brandão, Alexandre Santos*. (2024). Machine Learning for Unmanned Aerial Vehicles Navigation: An Overview. SN Computer Science. 5. 10.1007/s42979-023-02592-5

[6] Journal, IJSREM. (2023). Autonomous Drone Navigation. INTERANTIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT. 07. 1-11. 10.55041/ijsrem27050

[7] *Fadi AlMahamid, Katarina Grolinger*, Autonomous Unmanned Aerial Vehicle navigation using Reinforcement Learning: A systematic review, Engineering Applications of Artificial Intelligence, Volume 115, 2022, 105321, ISSN 0952-1976. 10.1016/j.engappai.2022.105321.

[8] *Ullah, Zaib & Al-Turjman, Fadi & Moatasim, Uzair & Mostarda, Leonardo & Gagliardi, Roberto.* (2020). UAVs joint optimization problems and machine learning to improve the 5G and Beyond communication. Computer Networks. 182. 10.1016/j.comnet.2020.107478

[9] *Choi, Su Yeon & Cha, Dowan*. (2019). Unmanned aerial vehicles using machine learning for autonomous flight; state-of-the-art. Advanced Robotics. 33. 1-13. 10.1080/01691864.2019.1586760

[10] *Quan, Yunhao & Cheng, Nan & Wang, Xiucheng & Shen, Jinglong & Ma, Longfei & Yin, Zhisheng.* (2023). Interpretable and Secure Trajectory Optimization for UAV-Assisted Communication. 10.48550/arXiv.2307.02002

[11] *Teixeira da Silva, Karolayne & Miguel, Geovane & Silva, Hugerles & Madeiro, Francisco*. (2023). A Survey on Applications of Unmanned Aerial Vehicles Using Machine Learning. IEEE Access. PP. 1-1. 10.1109/ACCESS.2023.3326101

[12] *Al-Haddad, Luttfi & Jaber, Alaa.* (2023). Applications of Machine Learning Techniques for Fault Diagnosis of UAVs. ceur-ws.org/Vol-3360/p03.pdf

[13] *Jacob, Billy & Kaushik, Abhishek & Velavan, Pankaj*. (2022). Autonomous Navigation of Drones Using Reinforcement Learning. 10.1007/978-981-16-7220-0_10

[14] *Yuzevych, V. M., Lozovan, V. P.* Influence of Mechanical Stresses on the Propagation of Corrosion Cracks in Pipeline Walls // Materials Science, 2022. Volume 57, No. 4. P. 539-548. 10.1007/s11003-022-00576-z

[15] *Сопрунюк П. М., Юзевич В. М.* Діагностика матеріалів і середовищ. Енергетичні характеристики поверхневих шарів. Львів: Сполом, 2005. 292 с. nvd-nanu.org.ua/d91f5656-e1ea-1bc9-3bfb-8d57729fbdad

[16] Дослідження взаємозв'язку між енергією поверхні волокнистих наповнювачів та міцністю полімерних композицій на їх основі / О. В. Миронюк, В. А. Дудко, Д. В. Баклан, К. О. Смольнич // Вісник НТУ "ХПІ". № 50 (1222). 2016. С. 3-8. mtsc.khpi.edu.ua/article/view/99759/95070

USING REINFORCEMENT LEARNING ALGORITHMS FOR UAV FLIGHT OPTIMIZATION

O. Dutsiak, V. Yuzevych

Ivan Franko National University of Lviv,
50 Drahomanova St., UA–79005 Lviv, Ukraine
oleg.dystak@gmail.com, yuzevych@ukr.net

Unmanned aerial vehicles (UAVs) have become a compelling research topic, offering potential for both civilian and military applications such as inspection, delivery, reconnaissance, and surveillance. Areas of interest include autonomous navigation, real-time path planning and object recognition. However, autonomous flight poses a number of challenges for UAVs, including the development of control strategies, parameter tuning, adaptive control, real-time trajectory planning, and object recognition in uncertain environments. In recent years, the implementation of machine learning has gained popularity as a promising approach to solving these problems in autonomous UAV flight. Machine learning gives UAVs the ability to discern patterns and make predictions based on data, thus bypassing the need for pre-programmed instructions during autonomous flight.

In this work, the main attention was focused on the study of a sub-method of machine learning, which is called learning with emphasis, with a view to optimizing the flight of a UAV by optimizing the process of constructing its path. A UAV trajectory planning method based on the gradient RL algorithm was proposed. It systematically presents the principles and components of the teaching method with emphasis. The path planning task is implemented using deep reinforcement learning, and the 3D environment is described using raster methods to construct a raster map. The study of the results of the functionality of the proposed algorithm was carried out in the environment of three-dimensional modeling. The algorithm is used to determine the optimal policy by treating each raster as a state, the 26 possible directions as action sets, and the action value function as the objective function.

When studying the results of the conducted experiment, it was found that after 160 experiments, thanks to the use of the RL gradient method, it was possible to determine the path from the starting point to the final point and successfully overcome all obstacles within a 84 seconds time period. Prospects for further improvement of the proposed method may include the study of methods to ensure more flexible movement within the map and optimization of sampling efficiency for improved real-world applicability.

Keywords: autonomous systems, machine learning, flight navigation, path planning, reinforcement learning, control algorithms.

Стаття надійшла до редакції 22.07.2024.
Прийнята до друку 05.08.2024.