УДК 681.3 https://doi.org/10.15407/pp2025.04.053

І.П. Рамик, Я.М. Ліндер

МЕТОДИ РЕАЛІЗАЦІЇ АВТОНОМНОСТІ КВАДРОКОПТЕРІВ НА ОСНОВІ ГІБРИДНИХ МЕТОДІВ НАВЧАННЯ

У роботі здійснено огляд та представлено аналіз методів реалізації автономності квадрокоптерів. Показано недоліки та обмеження класичного конвеєрного підходу «Сприйняття-Планування-Керування». Одним з його фундаментальних обмежень є нездатність математичних моделей врахувати всі складні ефекти непередбачуваного середовища. Натомість застосування алгоритмів машинного навчання дозволяє реалізовувати агентів керування на основі досвіду взаємодії агента з реальним чи симульованим середовищем. Це значно покращує адаптивність системи до нестандартних умов. Основна частина роботи присвячена порівнянню методів машинного навчання, що застосовувались до задачі реалізації автономності квадрокоптерів. Детально розглянуто методи навчання з підкріпленням. Зокрема, показано, що алгоритми, які не використовують модель світу, здатні перевершувати професіних пілотів в окремих задачах. Проте вони потребують значних обсягів даних та часу для навчання. Натомість навчання з підкріпленням на основі моделей підвищує ефективність тренування. В процесі тренування агент вивчає модель світу, що дозволяє йому передбачати динаміку середовища. Окремо в статті було розглянуто імітаційне навчання та похідні від нього. Ефективним підходом є послідовне застосування імітаційного навчання та навчання з підкріпленням, що дозволяє поєднувати їхні переваги. Було також розглянуто дослідження, що спираються на фізично інформовані методи, які базуються на використанні диференційованих симуляторів. Диференційовані симулятори дозволяють обчислювати градієнти функції втрат відносно параметрів керування. Всі розглянуті методи було проаналізовано в розрізі ефективності використання даних, вимог до обчислювальних ресурсів та фундаментальних обмежень. Результати аналізу можуть бути використані для вибору архітектури системи керування квадрокоптером залежно від доступних обчислювальних ресурсів та специфіки конкретного завдання.

Ключові слова: квадрокоптери, автономний політ, машинне навчання, навчання з підкріпленням, імітаційне навчання, бортовий комп'ютер, польотний контролер, диференційована симуляція.

I.P. Ramyk, Ya.M. Linder

METHODS FOR IMPLEMENTING QUADROTORS AUTONOMY BASED ON HYBRID LEARNING METHODS

This paper reviews and analyzes methods for achieving quadcopter autonomy. It shows disadvantages and limitations of the classical "Perception-Planning-Control" pipeline. A fundamental limitation of this approach is the inability of mathematical models to take into account all complex effects of the unpredictable environment. In return, the application of machine learning algorithms enables the implementation of control agents based on experience of interactions with real or simulated environments, significantly improving system adaptability to non-standard conditions. The core of this work compares machine learning methods applied to quadcopter autonomy task. It provides a detailed overview of reinforcement learning. It is shown that model-free algorithms are able to outperform professional human pilots in specific tasks. However, they require significant amounts of data and training time. In return, model-based reinforcement learning improves training efficiency. During the training, the agent learns a world model that can be used to predict environment dynamics. The article also explores imitation learning and derived methods. An effective approach is to sequentially apply imitation learning and reinforcement learning, which combines the strengths of both approaches. The paper reviews works relying on physics-informed methods using differentiable simulators. Differentiable simulators are used to calculate loss function gradients relative to control parameters. All discussed methods are analyzed regarding data efficiency, computational resource requirements, and fundamental limitations. The analysis results can be used to select quadcopter control architectures based on available computational resources and specific task requirements.

Keywords: quadrotors, autonomous flight, machine learning, reinforcement learning, imitation learning, companion computer, flight controller, differentiable simulation.

Вступ

Системи керування квадрокоптерами удосконалюються, перетворюючись на складні системи, здатні самостійно виконувати комплексні задачі без втручання людини.

Класичні методи високорівневої автоматизації польоту реалізують конвеєр “Сприйняття-Планування-Керування”, де окремі компоненти виконують ізольовані задачі на основі математичних моделей. Цей підхід забезпечує на виході передбачувану модель, проте вона не може врахувати всі аспекти складного середовища.

Методи машинного навчання досягають автономності інакше, а саме шляхом навчання стратегій керування через досвід. Такі методи дозволяють системам керування вдосконалюватись безпосередньо з даних польотів (симульованих або реальних).

Ця стаття розглядає апаратну архітектуру квадрокоптерів, а також методи реалізації їхньої автономності від класичних підходів до сучасних алгоритмів машинного навчання. В роботі розглянуто їхні переваги та обмеження.

Апаратна архітектура квадрокоптера

У дослідженнях, присвячених автономним квадрокоптерам, обчислювальні задачі зазвичай розподіляються між двома компонентами: низькорівневим польотним контролером і високорівневим бортовим комп’ютером [1].

Польотний контролер – це плата, яка відповідає за стабілізацію та безпеку польоту. Основною функцією польотного контролера є підтримання стабільності польоту, зокрема, утримання заданих кута і висоти. Польотний контролер зазвичай має інерційний вимірювальний пристрій (Inertial measurement unit, IMU) та, можливо, інші сенсори для визначення швидкості та орієнтації квадрокоптера у просторі. Він формує високочастотні сигнали для електронних регуляторів швидкості, таким чином впливаючи на тягу моторів, і, відповідно, рух квадрокоптера [3]. Польотний контролер може отримувати високорівневі команди (наприклад бажану орієнтацію чи швидкість) від пульта пілота або від бортового комп’ютера, який забезпечує автономність квадрокоптера (Рис. 1). Найбільш поширені вбудовані програми польотних контролерів включають PX4, ArduPilot або BetaFlight.

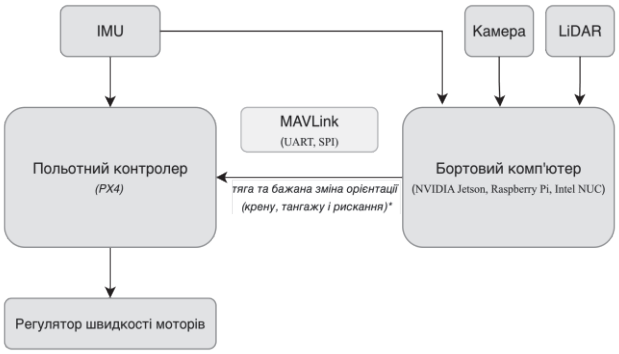

Рис. 1. Апаратна архітектура квадрокоптера

Бортовий комп’ютер – це окремий комп’ютер або модуль на борту квадрокоптера, що виконує ресурсомісткі задачі вищого рівня: обробку зображень, побудову карти середовища, планування траєкторії, ухвалення рішень тощо [3]. Бортовий комп’ютер отримує дані від додаткових сенсорів (як-от, камери, лідару) та від польотного контролера (телеметрія IMU, стан батареї). Комбінуючи всю наявну інформацію, він ухвалює рішення про подальші команди, які необхідно передати польотному контролеру.

Зв’язок між двома компонентами зазвичай здійснюється через високошвидкісний послідовний інтерфейс (UART, SPI) або Ethernet. Найчастіше протоколом обміну інформацією обирають MAVLink [3].

Така модульна архітектура доволі гнучка, адже польотний контролер здатний стабілізувати політ (або реалізовувати інший, визначений на цей випадок, план дій) навіть у ситуації, якщо бортовий комп’ютер виходить з ладу.

Розподіл задач між бортовим комп'ютером та польотним контролером став за-

гальноприйнятим та використовується в більшості досліджень.

Найбільш поширена конфігурація поєднує польотний контролер Pixhawk та бортовий комп'ютер NVIDIA Jetson. Завдяки графічному процесору NVDIA стає можливою реалізація алгоритмів комп'ютерного зору та інші складні обчислення. Наприклад, у роботі [4] така комбінація була використана для розробки систем пошуку та виявлення людей на відео з бортової камери. Аналогічні конфігурації було застосовано у роботі [5] для проходження смуг перешкод.

З інших бортових комп'ютерів у дослідженнях зустрічається використання моделей Intel, які також дозволяють обробляти великі обсяги даних у реальному часі [6].

Коли обчислення потребують менших потужностей, то можуть використовуватись одноплатні комп'ютери Raspberry Pi, які є більш енергоефективними та водночас дешевшими. Системи, реалізовані на Raspberry Pi, здатні забезпечувати просте розпізнавання зображення в режимі реального часу [7].

Класичні методи реалізації автономності квадрокоптерів

За класичним підходом до реалізації автономності квадрокоптерів будується конвеєр "Сприйняття-Планування-Керування", де виділяються три відповідні підзадачі, які реалізуються окремими компонентами. "Сприйняття" будує модель світу, "Планування" відповідає за прокладання маршруту, а "Керування" забезпечує відповідність цьому маршруту. Таке рішення забезпечує простоту розробки й інтерпретованість поведінки квадрокоптера. Проте воно має й ряд недоліків, зокрема відсутність зворотного зв'язку між компонентами та накопичення похибки на всіх етапах конвеєра [8].

"Сприйняття" покладається на сигнал з відео чи тепловізійної камери, GPS сигнал та бортові датчики для оцінки власного положення: орієнтації в просторі, швидкості тощо. Візуальна інерціальна одометрія (VIO) є підсистемою "Сприйняття", що поєднує дані з камер та інерційних блоків. Класичні методи VIO неефективні в специфічних умовах, таких як погане освітлення, політ над однотонною місцевістю [9].

"Планування" складається з двох етапів: спочатку здійснюється пошук шляху з урахуванням перешкод, а потім генерація траєкторії. Шлях являє собою послідовність точок. На його основі будується гладка траєкторія зазвичай у вигляді поліноміальних сплайнів [10].

Модуль "Керування" забезпечує дотримання цієї запланованої траєкторії шляхом надсилання сигналів електродвигунам. Зазвичай до цього процесу залучений високорівневий контролер, що регулює положення квадрокоптера в просторі та низькорівневий контролер орієнтації [11].

Найбільш поширеними є два типи контролерів: Пропорційно-інтегрально-диференціальні (ПІД) контролери та Лінійно-квадратичні регулятори. ПІД-контролери відносно прості та поширені, але їхньої ефективності недостатньо для керування динамікою квадрокоптера у високошвидкісних режимах [12]. Лінійно-квадратичний регулятор реалізує техніку оптимального керування, що забезпечує надійнішу та стабільну роботу, мінімізуючи функцію вартості помилок стану та керувальних впливів [12].

Проте класичні контролери покладаються на спрощені математичні моделі, які не можуть врахувати складні явища реального світу. Через це будь-яке отримане у такий спосіб керування є субоптимальним [12].

Саме ці обмеження стимулювали спроби реалізації автономності на основі навчання. Машинне навчання створює стратегію керування квадрокоптером шляхом спроб та помилок, здійснених в симульованому або реальному середовищах.

Нижче розглядаються основні підходи, що використовують машинне навчання для реалізації автономності квадрокоптерів.

Навчання з підкріпленням

Навчання з підкріпленням (Reinforcement Learning, RL) – це галузь машинного навчання для розв'язання задач Марковського процесу ухвалення рішень (МП). Ключові поняття RL – це агент та середовище. Агент навчається оптимальної стратегії взаємодії з середовищем. Середовище забезпечує зворотний зв'язок у вигляді винагород. Задачею агента є максимізація кумулятивної винагороди [13].

Задачу керування квадрокоптером можна формалізувати як МП, що визначається кортежем (S,A,P,r,γ). Тут S – простір станів (наприклад, положення, орієнтація, швидкості квадрокоптера), A – простір дій (команди для моторів або польотного контролера), $P(s_{t+1}|s_t,a_t)$ – функція ймовірності переходу до стану s_{t+1} зі стану s_t при виконанні дії a_t, $r(t)$ – миттєва винагорода, а $\gamma \in [0,1)$ – коефіцієнт знецінення, що визначає цінність майбутніх винагород [13]. Залежно від поставленої задачі, винагорода може визначатись по-різному, проте зазвичай агент, що керує квадрокоптером, отримує від'ємні винагороди за зіткнення, додатні за досягнення поставлених цілей, та невеликі нагороди на кожному кроці, які вказують на наближення чи віддалення від мети (наприклад, від'ємні винагороди, що за величиною пропорційні відстані до наступної цілі).

Метою агента є знаходження оптимальної стратегії π^*, яка максимізує очікувану кумулятивну дисконтовану винагороду:

$$\pi^* = \underset{\pi}{\operatorname{argmax}} E_{\tau \sim \pi}\left[\sum_{t=0}^{\infty} \gamma^t r(t)\right]$$

де τ – траєкторія, згенерована стратегією π [13].

Навчання з підкріпленням без моделі. Методи RL без моделі навчають стратегії π безпосередньо з досвіду, не намагаючись вивчити динаміку середовища [14]. Ці методи здатні досягати високої ефективності, що було показано в практичних експериментах, проте їхнім головним недоліком є потреба у великій кількості даних для навчання.

Одним із найпереконливіших прикладів застосування навчання з підкріпленням є система Swift [15], яка змогла перемогти чемпіонів світу з перегонів квадрокоптерів у змаганнях. Причому змагання відбувались на трасі, на якій квадрокоптер не літав до того. Система Swift використовувала алгоритм PPO (Proximal policy optimization) для тренування стратегії керування в симуляції. Через стан середовища агент отримував інформацію про позицію, швидкість, орієнтацію квадрокоптера, відносне розташування брами та свою попередню дію. Дія агента визначалась четвіркою чисел: колективною тягою та кутовими швидкостями корпусу квадрокоптера вздовж трьох осей. Ці команди далі обробляв польотний контролер. Агент отримував винагороду за наближення до центру наступної брами, а також за те, що тримав її в полі зору. За колізії та занадто різкі маневри отримував штрафи (від'ємні винагороди).

Інші дослідження також демонструють успішне застосування PPO для агресивного польоту в рандомізованих симуляційних сценаріях [16] , а також для керування роями квадрокоптерів у задачах спостереження [17].

Однак головним недоліком методів навчання з підкріпленням без моделі включно з PPO, є їхня низька ефективність використання даних. Вони вимагають великої кількості взаємодій із середовищем, особливо коли вхідні дані мають високу розмірність як, наприклад, необроблені зображення з камери.

Навчання з підкріпленням на основі моделі. На відміну від агента RL без моделі, агент навчання з підкріпленням на основі моделі (model-based RL, MBRL) не сприймає середовище як чорну скриню. Натомість він вивчає модель динаміки середовища. Цю модель називають "Моделлю світу". На практиці це призводить до зменшення кількості даних, необхідних для навчання агента [18]. Вивчивши модель світу, агент може використовувати її для прогнозування майбутніх сценаріїв та тренувати свою стратегію за допомогою цього прогнозування, не потребуючи такої великої кількості взаємодій із середовищем.

Наразі DreamerV3 є одним із найбільш розповсюджених алгоритмів MBRL, який продемонстрував переконливі результати в задачах керування [18]. Архітектура DreamerV3 складається з трьох ключових компонентів, які тренуються паралельно.

Першим компонентом є "Модель світу", яка відповідає за перетворення вхідних сенсорних даних високої розмірності в латентний стан. Модель світу також вчиться прогнозувати перехід з поточного латентного стану до наступного внаслідок обраної дії. Таким чином, модель вивчає та орієнтується на значущі ознаки середовища для моделювання сценаріїв [18].

Другий компонент, "Критик", навчається апроксимувати функцію цінності, яка визначає очікувану сумарну винагороду для траєкторій у латентному просторі вивченої моделі світу.

Третім компонентом є "Актор" (Actor, виконавець). Компонент "Актора" безпосередньо навчає стратегії керування. Під час навчання "Актор" максимізує оцінки цінності, що надаються "Критиком" для траєкторій, згенерованих у латентному просторі "Моделі світу".

DreamerV3 був використаний для тренування квадрокоптера, що взяв участь в перегонах. Модель навчалась безпосередньо на необроблених пікселях з бортової камери [18]. Дослідження показало що DreamerV3 успішно впорався із завданням, тоді як алгоритм PPO не зміг навчитися керувати квадрокоптером, отримуючи ті самі дані для навчання (необроблені зображення) [18].

Імітаційне навчання

В основі Імітаційного навчання (Imitation Learning, IL) лежить ідея вивчення стратегії, яка імітує дії експерта [21]. На відміну від RL, де агент досліджує середовище шляхом спроб та помилок, в IL агенту показують набір демонстрацій експерта [22].

Демонстрації можуть бути надані пілотом квадрокоптера або згенеровані контролером. Тож IL можна розглядати як кероване навчання.

Класичне IL, або клонування поведінки, має суттєві недоліки, що прямо випливають з особливостей навчання. Проблеми виникають, коли навчена стратегія потрапляє у стани, яких не було в демонстраційній вибірці. Оскільки вона не отримала даних про еталонну поведінку в таких станах, навіть невеликі помилки можуть накопичуватися, призводячи до непередбачуваної поведінки. Поширеною спробою розв'язання цієї проблеми є агрегація набору даних (Dataset Aggregation, DAgger), що збирає дані в станах, які відвідує агент згідно навченої стратегії, і запитує в експерта правильні дії [20].

Ще одним суттєвим недоліком IL є те, що через особливості навчання, отримана стратегія обмежена продуктивністю експерта. Вона може навчитися імітувати експерта, але не перевершити його [20].

Привілейоване навчання

Привілейоване навчання можна розглядати як логічний крок у розвитку IL. В симуляційному середовищі можливо створити експерта, який володіє повною інформацією про його стан, що, очевидно, не доступно квадрокоптеру під час звичайного польоту. Такого експерта можна використати для тренування стратегії на основі IL, що вчиться відтворювати еталонні дії не маючи доступу до всієї інформації. Стратегія, що вчиться, отримує на вхід лише реалістичні, зашумлені сенсорні дані (наприклад, зображення з камери, дані IMU), до яких квадрокоптер і буде мати доступ в реальному світі [21].

У дослідженні [21] стратегія керування була повністю навчена в симуляції за допомогою привілейованого оптимального контролера, який мав доступ до повної інформації про середовище в симуляторі. Квадрокоптер зміг виконати складні маневри, зокрема, повний оберт у повітрі, під час яких досягав прискорення до 3g.

Переконливим результатом є також те, що стратегія керування не потребувала додаткового навчання в даних з реального світу, перехід від симуляції відбувся без ускладнень [21].

Гібридні методології

На практиці дослідники часто використовують IL та RL послідовно для отримання кращих результатів [22].

Спочатку стратегія навчається за допомогою IL на демонстраційному наборі, наданому експертом. Результатом цього етапу є доволі ефективна початкова стратегія, яку потім покращує алгоритм RL. Таким чином IL усуває найбільшу проблему RL, а саме неефективну початкову фазу навчання, під час якої агент діє майже випадково. Водночас застосування RL дозволяє перевершити стратегію експерта [22].

Залишкове навчання з підкріпленням. Ідея залишкового навчання з підкріпленням полягає в тому, щоб поєднати переваги класичних ПІД-контролерів з можливістю врахування складних явищ реального світу, яке забезпечує навчання з підкріпленням.

У такій системі основою керування є класичний контролер. Модель навчання з підкріпленням відповідає за компенсацію динаміки, що не передбачено у випадку розробки контролера. Під час тренування агент вчиться, яку коригувальну дію потрібно додати до сигналу контролера в різних умовах [24].

Висока точність та адаптивність цього методу була підтверджена у дослідженні [25].

Фізично-інформовані методи навчання

В цьому розділі розглядаються методи, які враховують знання про фізичні процеси безпосередньо під час навчання. Ключовою технологією, яка використовується в цих методах, є диференційовані симулятори. На відміну від традиційних симуляторів, які є чорними скриньками для алгоритму тренування, диференційовані симулятори дозволяють обчислювати градієнти першого порядку від цільової функції, наприклад, помилки траєкторії, і використовувати це в процесі тренування для оновлення стратегії. Градієнти першого порядку мають нижчу дисперсію, ніж стохастичні оцінки, що використовуються в алгоритмах навчання з підкріпленням, на кшталт PPO. Це забезпечує швидше тренування моделі [26, 27, 28].

Нижче описано процес тренування у диференційованому симуляторі, що використовувався у [26].

Система тренує модель динаміки квадрокоптера, починаючи з простої аналітичної моделі [26, 29]. А також безперервно збирає дані під час польоту квадрокоптера і використовує їх для тренування залишкової моделі, яка має компенсувати ефекти, що не закладались у базову модель. Це може бути аеродинамічний опір, пориви вітру, будь-які інші явища реального світу, які складно завчасно аналітично представити.

Інтеграція оновленої моделі динаміки (включно з навченою залишковою моделлю) в диференційований симулятор дає змогу виконувати наскрізне диференціювання. Це дозволяє обчислювати градієнти цільової функції (наприклад, помилки траєкторії), диференціюючи її щодо параметрів стратегії крізь весь процес симуляції, і використовувати їх для прямої оптимізації [26].

Використання градієнтів на основі диференційованої симуляції виявляється ефективнішим за використання градієнтів нульового порядку в PPO.

У дослідженні [26] порівняли агентів навчених алгоритмом на основі диференційованої симуляції та RL на задачі зависання. Було показано, що диференційована симуляція забезпечує кращу стійкість до збурень, тоді як PPO потребує значно більше симуляційних кроків і гірше адаптується до змінних умов.

Порівняння

У даній роботі було розглянуто основні методи реалізації автономності квадрокоптерів. Кожен метод має свої переваги, вимоги до середовища та даних. Ці особливості визначають, який із методів найкращі для конкретної задачі. У Табл. 1 наведено порівняльний аналіз розглянутих методів.

Таблиця 1.

Порівняльний аналіз методів навчання автономних квадрокоптерів

Метод	Вимоги до даних та/або середовища	Переваги	Недоліки
RL (без моделі) [15]	Потребує великої кількості взаємодій з середовищем.	Здатний досягати ефективності, що переважає рівень професійних пілотів.	Низька ефективність даних.
RL (на основі моделі) [18]	Потребує середовища для симуляцій.	Висока ефективність даних. Здатний досягати ефективності, що переважає рівень експертів.	Велика обчислювальна складність тренування та роботи в реальному часі.
Імітаційне навчання (IL) [20]	Потребує демонстрацій від експерта (наприклад, професійного пілота).	Швидке навчання завдяки демонастраційній вибірці.	Потреба демонстрацій від експерта. Ефективність обмежена ефективністю експерта.
Гібридний: IL + RL [22, 23]	Потребує демонстрацій від експерта (наприклад, професійного пілота).	Дозволяє перевершити експерта (на відміну від чистого IL). Краща ефективність даних, ніж в RL.	Потреба демонстрацій від експерта.
Привілейоване навчання [21]	Потребує привілейованого експерта, що має доступ до повної інформації про стан середовища.	Дозволяє вивчати оптимальні дії для умов складної/незвичної динаміки.	Необхідність привілейованого експерта.
Залишкове RL (Residual RL) [24, 25]	Потребує наявності стабільного базового класичного контролера та значної кількості даних.	Підвищує точність класичних контролерів.	Залежність від ефективності класичного контролера.
Фізично-інформовані методи навчання [26, 29]	Вимагає диференційованого симулятора.	Висока ефективність даних. Демонструє значно швидше навчання, ніж RL.	Необхідність мати повністю диференційовану модель динаміки.

Одним із ключових критеріїв оцінки методів машинного навчання є ефективність використання даних. Як видно з таблиці 1, навчання з підкріпленням без моделі [15], хоч і здатне перевершувати професійних пілотів, має доволі низьку

ефективність даних, що вимагає тривалого тренування.

Цю проблему вирішують методи, що враховують динаміку середовища під час тренування: RL на основі моделі [18] та фізично інформовані методи [26, 29].

Висновки

Методи машинного навчання, зокрема, навчання з підкріпленням, мають такі переваги:

1. кращу адаптивність до складної, змінної динаміки, ніж класичні методи на основі спрощених математичних моделей;
2. можливість перевершувати рівень професійних пілотів;
3. наявність різних підходів дозволяє обирати метод, оптимальний для конкретного класу задач і умов експлуатації.

Проведений порівняльний аналіз виявив прогалини в існуючих алгоритмах, а саме:

1. для деяких алгоритмів актуальна проблема неузгодженості між цифровою моделлю, на якій тренується агент, і реальним світом;
2. низька ефективність даних для деяких існуючих алгоритмів;
3. низька здатність до узагальнення та проблема виродження стратегій.

Проведений порівняльний аналіз стане основою для вибору ефективного алгоритму машинного навчання у задачі автономної навігації квадрокоптера у міській забудові.

Література

1. Lukash Y., Prystavka P. A research platform for vision-based UAV autonomy: Architecture and implementation. Fourth International Conference on Cyber Hygiene & Conflict Management in Global Information Networks (CHCM 2025). 2025. Vol. 4024. P. 250-259. URL: https://ceur-ws.org/Vol-4024/paper16.pdf (дата звернення: 05.11.2025).
2. Faessler M., Fontana F., Forster C., Scaramuzza D. Automatic Re-Initialization and Failure Recovery for Aggressive Flight with a Monocular Vision-Based Quadrotor. 2015 IEEE International Conference on Robotics and Automation (ICRA). 2015. P. 1722–1729. DOI: 10.1109/ICRA.2015.7139420.
3. Companion Computers | PX4 Guide (main). PX4 Documentation. URL: https://docs.px4.io/main/en/companion_computer/ (дата звернення: 05.11.2025).
4. Ciccone F., Ceruti A. Real-Time Search and Rescue with Drones: A Deep Learning Approach for Small-Object Detection Based on YOLO. Drones. 2025. Vol. 9, no. 8. P. 514. DOI: 10.3390/drones9080514.
5. Jain R., Jones C., Lucas R., Siddiqui H. Autonomous Aerial Drone with Infrared Depth Tracking. University of Central Florida. 2020. URL: https://www.ece.ucf.edu/seniordesign/fa2019sp2020/g18/G18_Conference%20Paper.pdf (date of access: 05.11.2025).
6. Liu X., Nardari G. V., Cladera Ojeda F., Tao Y., Zhou A., Donnelly T., Qu C., Chen S. W., Romero R. A. F., Taylor C. J., Kumar V. Large-scale Autonomous Flight with Real-time Semantic SLAM under Dense Forest Canopy. arXiv. 2021. DOI: 10.48550/arXiv.2109.06479.
7. Daspan A., Nimsongprasert A., Srichai P., Wiengchand P. Implementation of Robot Operating System in Raspberry Pi 4 for Autonomous Landing Quadrotor on ArUco Marker. International Journal of Mechanical Engineering and Robotics Research. 2023. Vol. 12, no. 4. P. 210–217. URL: https://www.ijmerr.com/2023/IJMERR-V12N4-210.pdf (date of access: 05.11.2025).
8. Xiao J., Zhang R., Zhang Y., Feroskhan M. Vision-based Learning for Drones: A Survey. arXiv preprint. 2023. arXiv:2312.05019. DOI: 10.48550/arXiv.2312.05019.
9. George A., Koivumäki N., Hakala T., Suomalainen J., Honkavaara E. Visual-Inertial Odometry Using High Flying Altitude Drone Datasets. Drones. 2023. Vol. 7, no. 1. C. 36. DOI: 10.3390/drones7010036.
10. Richter C., Bry A., Roy N. Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments. In: Masayuki Inaba, Peter Corke (eds.) Robotics Research. The 16th International Symposium ISRR, 16–19 December 2013, Singapore. Springer Tracts in Advanced Robotics, vol. 114. Cham: Springer, 2016. C. 649–666. DOI: 10.1007/978-3-319-28872-7_37.

11. Mamo M. B. Trajectory tracking control of quadcopter by designing Third order SMC Controller. Global Scientific Journals. 2020. Vol. 8, no. 9. C. 2100–2108.
12. Abu Ihnak M. S., Edardar M. M. Comparing LQR and PID Controllers for Quadcopter Control Effectiveness and Cost Analysis. 2023 International Conference on Systems and Control (ICSC). IEEE, 2023. C. 770–775. DOI: 10.1109/ICSC58660.2023.10449763.
13. Mnih V., Kavukcuoglu K., Silver D., Rusu A. A., Veness J., Bellemare M. G., Graves A., Riedmiller M., Fidjeland A. K., Ostrovski G., Petersen S., Beattie C., Sadik A., Antonoglou I., King H., Kumaran D., Wierstra D., Legg S., Hassabis D. Human-level control through deep reinforcement learning. Nature. 2015. Vol. 518, no. 7540. C. 529–533. DOI: 10.1038/nature14236.
14. Olivares D., Fournier P., Vasishta P., Marzat J. Model-Free versus Model-Based Reinforcement Learning for Fixed-Wing UAV Attitude Control Under Varying Wind Conditions. arXiv preprint. 2024. arXiv:2409.17896. DOI: 10.48550/arXiv.2409.17896.
15. Kaufmann E., Bauersfeld L., Loquercio A., Müller M., Koltun V., Scaramuzza D. Champion-level drone racing using deep reinforcement learning. Nature. 2023. Vol. 620, no. 7976. C. 982–987. DOI: 10.1038/s41586-023-06419-4.
16. Mengozzi S. Learning Agile Flight Using Massively Parallel Deep Reinforcement Learning: Master's thesis. University of Bologna, Department of Electrical, Electronic, and Information Engineering "Guglielmo Marconi", 2022. 67 p. URL: https://amslaurea.unibo.it/id/eprint/28648/1/SebastianoMengozzi_Thesis.pdf (дата звернення: 11.11.2025).
17. Arranz R., Carramiñana D., de Miguel G., Besada J. A., Bernardos A. M. Application of Deep Reinforcement Learning to UAV Swarming for Ground Surveillance. Sensors. 2023. Vol. 23, no. 21. C. 8766. DOI: 10.3390/s23218766.
18. Romero A., Shenai A., Geles I., Aljalbout E., Scaramuzza D. Dream to Fly: Model-Based Reinforcement Learning for Vision-Based Drone Flight. arXiv preprint. 2025. arXiv:2501.14377. DOI: 10.48550/arXiv.2501.14377.
19. Chen D., Zhou B., Koltun V., Krähenbühl P. Learning by Cheating. arXiv preprint. 2019. arXiv:1912.12294. DOI: 10.48550/arXiv.1912.12294.
20. Pfeiffer C., Wengeler S., Loquercio A., Scaramuzza D. Visual Attention Prediction Improves Performance of Autonomous Drone Racing Agents. arXiv preprint. 2022. arXiv:2201.02569. DOI: 10.48550/arXiv.2201.02569.
21. Kaufmann E., Loquercio A., Ranftl R., Müller M., Koltun V., Scaramuzza D. Deep Drone Acrobatics. Robotics: Science and Systems (RSS), 2020. DOI: 10.48550/arXiv.2006.05768.
22. Abusadeh R. Evaluation of Imitation Learning with Reinforcement Learning-Based Fine-Tuning for Different Control Tasks. Master's thesis. Czech Technical University in Prague, Faculty of Electrical Engineering, Department of Cybernetics, 2025. URL: https://dspace.cvut.cz/bitstream/handle/10467/120386/F3-DP-2025-Abusadeh-Rawan-evaluation_IL_RL.pdf (дата звернення: 05.11.2025).
23. Xing J., Romero A., Bauersfeld L., Scaramuzza D. Bootstrapping Reinforcement Learning with Imitation for Vision-Based Agile Flight. arXiv preprint. 2024. arXiv:2403.12203. DOI: 10.48550/arXiv.2403.12203.
24. Nahrendra I. M. A., Tirtawardhana C., Yu B., Lee E. M., Myung H. Retro-RL: Reinforcing Nominal Controller With Deep Reinforcement Learning for Tilting-Rotor Drones. arXiv preprint. 2022. arXiv:2207.03124. DOI: 10.48550/arXiv.2207.03124.
25. Zhang R., Zhang D., Mueller M. ProxFly: Robust Control for Close Proximity Quadcopter Flight via Residual Reinforcement Learning. arXiv preprint. 2024. arXiv:2409.13193. DOI: 10.48550/arXiv.2409.13193.
26. Blukis V., Huber S., Wrona K. K., Büchler D., Muehlebach M., Krause A., Hanna J. P., Hennan D. D., Ansari S. S. P. Learning on the Fly: Rapid Policy Adaptation via Differentiable Simulation. arXiv preprint. 2025. arXiv:2508.21065. DOI: 10.48550/arXiv.2508.21065.
27. Schnell P., Thuerey N. Stabilizing Backpropagation Through Time to Learn Complex Physics. International Conference on Learning Representations (ICLR), 2024. arXiv:2405.02041. DOI: 10.48550/arXiv.2405.02041.
28. Ren J., Yu C., Chen S., Ma X., Pan L., Liu Z. DiffMimic: Efficient Motion Mimicking with

Differentiable Physics. arXiv preprint. 2023. arXiv:2304.03274. DOI: 10.48550/arXiv.2304.03274.
29. Heeg J., Song Y., Scaramuzza D. Learning Quadrotor Control From Visual Features Using Differentiable Simulation. arXiv preprint. 2024. arXiv:2410.15979. DOI: 10.48550/arXiv.2410.15979.

Одержано: 19.11.2025
Внутрішня рецензія отримана: 26.11.2025
Зовнішня рецензія отримана: 28.11.2025

Про авторів:

[1]Рамик Іван Петрович,
аспірант.
https://orcid.org/0009-0008-5034-676X

[1]Ліндер Ярослав Миколайович,
кандидат фізико-математичних наук,
доцент кафедри Інтелектуальних Програмних Систем.
https://orcid.org/0000-0003-1076-9211

Місце роботи авторів:

[1]Факультет комп'ютерних наук та кібернетики Київського національного університету імені Т.Г. Шевченка
тел. +38(044) 521-32-74
E-mail: csc@knu.ua
https://csc.knu.ua