Лабораторна робота №4. Рій із кількох апаратів: уникнення зіткнень

2 вересня 2026 р.

Мета

Перейти від одного апарата Crazyflie 2.1 до рою з кількох. Побудувати вектор спостереження з блоком сусідів, інваріантний до їх порядку та кількості. Навчити спільну політику без зіткнень і оцінити її на 50 епізодах та переносом на більший рій.

Склад бригади і розподіл ролей

Робота виконується бригадою з трьох осіб. Кожен учасник відповідає за свою підсистему й захищає її окремо.

- **Учасник A — середовище рою.** Розширює середовище з роботи №3 на кілька апаратів, реалізує базову політику без урахування сусідів і лічильник зіткнень.
- **Учасник B — інваріантний кодувальник.** Формує вектор спостереження з блоком сусідів, реалізує агрегатор і доводить його інваріантність до перестановки, а потім ламає її навмисне.
- **Учасник C — навчання і перенос.** Навчає політику зі спільними параметрами, знімає статистику на 50 епізодах і запускає навчену політику на вдвічі більшому рої.

Інтеграційний крок виконують разом: кодувальник учасника B під'єднується до середовища учасника A, і на цій збірці учасник C проводить навчання та перенос.

Теоретичні відомості

У роботі №3 апарат бачив 18 чисел про себе: 3 на вектор до цілі, 3 на лінійну швидкість, 9 на матрицю повороту й 3 на кутову швидкість. У рої додається блок про K найближчих сусідів, а не про всіх. На кожного беруть різницю позицій Δp і швидкостей Δv, бо політика має працювати в будь-якій точці кімнати.

Блок	Компоненти	Розмір
Своє (як у роботі №3)	ціль, швидкість, орієнтація, кутова швидкість	18
На кожного сусіда	Δp (3) + Δv (3)	6
Разом при $K = 4$	$18 + 4 \times 6$	**42**

Просте склеювання станів усіх N апаратів ламається з двох причин.

Перша: розмірність росте разом із роєм. Склеювання ($N \times 18$) дає вхід 108 чисел для 6 апаратів, 180 для 10 і 2304 для 128. Блок сусідів ($18 + 6K$ при $K = 4$) дає 42 числа за будь-якого N.

Друга: перестановка сусідів змінює відповідь. Візьміть Δx чотирьох сусідів (0,80; $-0{,}50$; 0,10; $-1{,}60$ м) і ваги $w = (2{,}0;\ 0{,}5;\ -1{,}0;\ 1{,}5)$. Порядок «Д2, Д4, Д3, Д6» дає $-1{,}15$, а «Д3, Д6, Д2, Д4» дає $-2{,}15$ при тих самих місцях апаратів. Середнє ж тих самих чисел дорівнює $-0{,}30$ за будь-якого порядку, тому сусідів агрегують:

$$e = \sum_{j=1}^{K} \alpha_j \, f(\Delta p_j, \, \Delta v_j), \qquad \sum_{j=1}^{K} \alpha_j = 1 \tag{1}$$

де e — агрегат сусідів, його довжина не залежить ані від K, ані від N; f — спільний кодувальник: 6 чисел на вході, 8–16 на виході; Δp_j, Δv_j — відносні позиція та швидкість j-го сусіда; α_j — вага j-го сусіда. У схемі **Deep Sets** усі $\alpha_j = 1/K$ (усереднення), у схемі уваги (**attention**) мережа рахує їх сама: для сусідів на 0,81 / 0,95 / 1,41 / 1,68 м вона дає близько 0,55 / 0,22 / 0,15 / 0,08.

Корисні посилання для ознайомлення:

- Zaheer та ін. Deep Sets — архітектурний прийом, на якому побудований агрегатор;
- Batra та ін. Decentralized Control of Quadrotor Swarms — джерело чисел для цієї роботи;
- Hüttenrauch та ін. Deep RL for Swarm Systems — інваріантні подання для роїв;
- Preiss та ін. Crazyswarm: A Large Nano-Quadcopter Swarm — реальний рій із 49 апаратів.

Середовище

Python 3.12, `gym-pybullet-drones` (MIT) з усіма залежностями: `numpy`, `matplotlib`, `gymnasium`, `stable-baselines3`, `pybullet`. Усе встановлюється з PyPI і GitHub.

Одна тонкість встановлення: `pybullet` не має готових збірок під Python 3.12, тому компілюється з джерел і потребує компілятора C++. На Ubuntu це `sudo apt install build-essential`, на macOS — інструменти командного рядка Xcode. Якщо збірка не проходить, робіть роботу в хмарному ноутбуці з попередньо встановленим компілятором — умови завдання від цього не змінюються.

Порядок виконання

Обчислювальний бюджет. Відеокарта не потрібна: усі запуски розраховані на процесор звичайного ноутбука. Тримайтеся межі **30 хвилин на один запуск навчання**; кількість кроків добирайте під цю межу й фіксуйте її в ноутбуці однаковою для всіх порівнюваних запусків. Криву, що не встигла вийти на полицю, теж можна аналізувати — висновок робиться про *однаково* недонавчені політики.

1. **[A]** Розширити середовище з роботи №3 на кілька апаратів і реалізувати базову політику, яка веде кожен апарат на його ціль і сусідів не бачить. Додати облік зіткнень і мінімальної попарної відстані.

2. **[A]** Прогнати базову політику на розстановці, де траєкторії перетинаються, і зафіксувати, скільки пар зіткнулося.

3. **[B]** Сформувати вектор спостереження з блоком K найближчих сусідів. Показати, що його довжина не залежить від розміру рою, і подати склад вектора таблицею.

4. **[B]** Реалізувати агрегатор (1) й довести його інваріантність до порядку сусідів на кількох перестановках. Потім зламати інваріантність навмисне й показати, що перевірка це виявляє.

5. **[С]** Навчити спільну політику зі спільними параметрами, заклавши у винагороду плавний штраф за близькість. Побудувати криву винагороди й криву зіткнень.

6. **[С]** Порівняти базову й навчену політики на 50 епізодах за безпекою та часткою тих, що дійшли до цілі.

7. **[разом]** Запустити навчену політику на вдвічі більшому рої без перенавчання й зафіксувати деградацію числами. Пояснити, що перенеслося, а що ні.

8. **[разом] Оформити ноутбук.** Зібрати роботу в один Jupyter-ноутбук: код кожного учасника у своїх комірках із короткими коментарями там, де це доречно. Обов'язково всередині: склад вектора спостереження, вивід самоперевірки інваріантності, криві навчання й зіткнень, таблицю порівняння політик і числа деградації при переносі, а також висновки — три-п'ять тверджень із числами. Ноутбук має виконуватися від початку до кінця без ручних правок; у першій комірці вкажіть, хто з учасників за які комірки відповідав.

Контрольні запитання

1. Скільки чисел у спостереженні при $K = 4$ і чому воно не залежить від N?

2. Чому в блоці сусіда подають Δp і Δv, а не абсолютні позиції та швидкості?

3. Наведіть числовий приклад, де перестановка сусідів змінює дію політики.

4. Чому агрегація інваріантна до порядку сусідів, а конкатенація ні?

5. Чим attention відрізняється від усереднення і коли дасть виграш?

6. Навіщо сортувати сусідів за відстанню, якщо агрегатор інваріантний?

7. Чому штраф за близькість роблять плавним, а не бінарним?

8. Рій із шести апаратів відкрив 8 клітинок, і кожен отримав $+8$. У чому проблема розподілу заслуг і як її лагодить винагорода 0,7 / 0,3?

9. Чому для однорідного рою вигідніша одна спільна мережа, ніж N окремих?

10. Що змінилося у вхідних даних, коли політику з 4 апаратів запустили на 8?