

# РОЗРОБКА СЕРВІС-ОРІЄНТОВАНОЇ СИСТЕМИ КООРДИНАЦІЇ РОЮ ДРОНІВ НА ОСНОВІ АЛГОРИТМУ МУРАШИНИХ КОЛОНІЙ (АСО) ТА СТИГМЕРГІЙНОЇ ВЗАЄМОДІЇ

Будзинський А.В.<sup>1</sup>, Булах Б.В.

Національний технічний університет України  
“Київський політехнічний інститут імені Ігоря Сікорського”, Київ, Україна

<sup>1</sup> kergontop@gmail.com

У дослідженні розглядається сервіс-орієнтована система координації рою дронів, що поєднує алгоритм мурашиних колоній і стигмергійну взаємодію. Архітектуру реалізовано у вигляді ROS2-вузлів: побудова та випаровування феромонної карти, відкладення феромонів і АСО-прийняття рішень; виконано інтеграцію з Gazebo/ArduPilot. Досягнуто візуалізації карти та продемонстровано консольні рішення АСО. Новизною даного дослідження є сервісна декомпозиція і безцентрова координація, а практична цінністю - масштабованість і стійкість до втрат зв'язку.

**Ключові слова:** АСО, стигмергія, роєва координація, ROS2, феромонна карта, Gazebo/ArduPilot.

## 1. ВСТУП

Координація рою безпілотних літальних апаратів (БПЛА) у динамічних та частково спостережуваних середовищах є актуальною науково-прикладною задачею. Вона ускладнюється обмеженою пропускну здатністю каналів зв'язку, їх перервністю, невизначеністю сенсорних даних, потребою швидкого масштабування та стійкості до відмов окремих агентів. Класичні централізовані підходи потребують стабільних каналів, глобальної оптимізації та часто деградують при збільшенні кількості дронів або втраті зв'язку, що мотивує перехід до розподілених методів координації.

Метою даного дослідження є розробка та експериментальна верифікація сервіс-орієнтованої системи координації рою БПЛА на основі алгоритму мурашиних колоній (АСО) та стигмергійної взаємодії, яка забезпечує масштабоване, відмовостійке й малозалежне від якості зв'язку узгодження дій агентів у реальному часі.

У цьому дослідженні було розроблено сервіс-орієнтовану систему (Service-Oriented Architecture, SOA) для координації рою дронів, що поєднує АСО зі стигмергією – непрямою координацією через “сліди” в середовищі (феромони). Рішення реалізовано на ROS 2, інтегровано з Gazebo/ArduPilot та демонструє візуалізацію феромонної карти й процес прийняття рішень у реальному часі.

## 2. ТЕОРЕТИЧНА СКЛАДОВА АЛГОРИТМУ АСО

АСО моделює поведінку мурах, які залишають феромони на корисних маршрутах. Інші агенти, зчитуючи ці “сліди”, імовірно обирають подальші кроки. У загальному вигляді правило переходу (оцінка привабливості) з поточного стану  $i$  до кандидата  $j$  визначається зваженою комбінацією феромону  $\tau_{ij}$  та евристики  $\eta_{ij}$  [1]:

$$score_j = (\tau_{ij} + \epsilon)^\alpha * (\eta_{ij})^\beta, p_j = \frac{score_j}{\sum_{i=1}^k score_i}, \text{ де}$$

$\tau_{ij}$  – феромон (накопичений досвідом “якість” переходу  $i \rightarrow j$ ). Чим більше, тим привабливіше,  
 $\eta_{ij}$  – евристика (миттєва корисність без досвіду). У даній роботі я використовував обернену відстань, що віддає перевагу близьким крокам при рівних феромонах,  
 $\alpha \geq 0$  – вага феромону (ступінь експлуатації минулого досвіду),  
 $\beta \geq 0$  – вага евристики (ступінь жадібності до локально кращого кроку),  
 $\epsilon$  – незначне чисельне значення для уникнення нулів.

Після кожного кроку феромон оновлюється. Однією з найважливіших подій оновлення є випаровування феромону для уникнення локальних максимумів на карті, яке розраховується за формулою [2]:

$$\tau_{ij} = (1 - \rho) * \tau_{ij}, \text{ де}$$

$\tau_{ij}$  – значення феромону між кроками  $i$  та  $j$ ,  
 $\rho$  – швидкість “забування”. Чим більше це значення, тим швидше відбувається адаптація до змін, але зменшується пам’ять.

Як було зазначено вище, у якості евристики ( $\eta$ ) було використано обернену відстань, що віддає перевагу близьким крокам при рівних феромонах. В залежності від задачі евристика може кодувати [3]:

- очікуваний приріст інформації
- енергетичну вартість маршруту
- безпекові / колізійні ризики

Формула обраної евристики (обернена відстань) має наступний вигляд:

$$\eta_j = 1/(d(i, j) + \epsilon).$$

### 3. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Для реалізації поставленого завдання було розроблено декілька сервісів за допомогою мови програмування Python та бібліотеки ROS 2, серед яких:

- PheromoneMapNode – сервіс для отримання, застосування та зберігання феромонів на карті.
- PheromoneDepositNode – сервіс для періодичної публікації депозиту феромонів у поточній позиції.
- ACODecisionNode – сервіс ймовірнісного прийняття рішень на основі карти феромонів.
- ACONavigationBridge – сервіс передачі інформації з ACODecisionNode до дронів.
- ACOSVisualizer – утиліта візуалізації карти феромонів.

Після проведення перших тестових завдань мною була отримана та візуалізована карта феромонів, наведена на рис. 1.

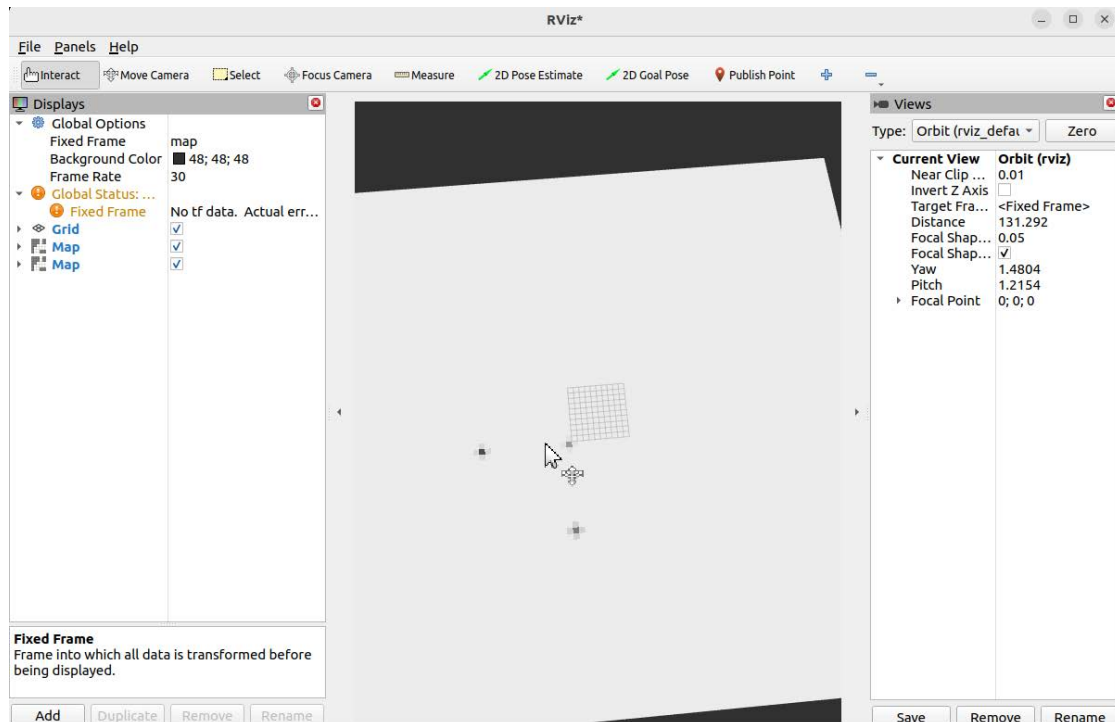


Рисунок 1. Приклад візуалізації карти феромонів

За умови ручного оновлення карти феромонів мною також був протестований створений мною сервіс прийняття рішень (ACODecisionNode) та логіка випаровування феромонів (рис. 2).

```
[INFO] [1763582030.551618187] [aco_decision_node]: From 4 → Next 2 (5.00, 5.00)
[INFO] [1763582031.592649563] [aco_decision_node]: From 2 → Next 4 (10.00, -5.00)
[INFO] [1763582032.550393293] [aco_decision_node]: From 4 → Next 2 (5.00, 5.00)
[INFO] [1763582033.546124116] [aco_decision_node]: From 2 → Next 4 (10.00, -5.00)
[INFO] [1763582034.576124551] [aco_decision_node]: From 4 → Next 2 (5.00, 5.00)
[INFO] [1763582035.545978100] [aco_decision_node]: From 2 → Next 4 (10.00, -5.00)
[INFO] [1763582036.554559491] [aco_decision_node]: From 4 → Next 2 (5.00, 5.00)
[INFO] [1763582037.546460651] [aco_decision_node]: From 2 → Next 4 (10.00, -5.00)
[INFO] [1763582038.592172615] [aco_decision_node]: From 4 → Next 1 (5.00, 0.00)
[INFO] [1763582039.554072164] [aco_decision_node]: From 1 → Next 4 (10.00, -5.00)
[INFO] [1763582040.567552825] [aco_decision_node]: From 4 → Next 0 (0.00, 0.00)
[INFO] [1763582041.556778630] [aco_decision_node]: From 0 → Next 4 (10.00, -5.00)
[INFO] [1763582042.553941176] [aco_decision_node]: From 4 → Next 0 (0.00, 0.00)
[INFO] [1763582043.556670735] [aco_decision_node]: From 0 → Next 4 (10.00, -5.00)
[INFO] [1763582044.550323224] [aco_decision_node]: From 4 → Next 3 (0.00, 5.00)
```

Рисунок 2. Приклад роботи алгоритму прийняття рішень

У даному прикладі мною було залишено 100 феромонів у точці 4 (10, -5), що автоматично зробило її більш привабливою за інші (на кожному етапі відбувається повернення

до точки 4). Від точки 4, де немає доступних пунктів призначення з високим рівнем феромонів, алгоритм за замовчуванням використовує евристику на основі відстані ( $\eta^\beta$ ), створюючи ймовірнісну поведінку локального дослідження. Поточні параметри ( $\alpha=1,0$ ,  $\beta=2,0$ ) створюють сильне слідування за феромонами після їх відкладення, але це не є хорошим вибором при рівномірному розподілі феромонів. Це демонструє важливість налаштування параметрів для бажаної поведінки рою. Гаусівська функція розповсюдження створює реалістичні градієнти феромонів, причому сусідні комірки отримують  $\sim 13,6\%$  від сили відкладення, що забезпечує більш плавні поля притягання феромонів.

#### 4. ПОДАЛЬШІ ДОСЛІДЖЕННЯ

Основним напрямком подальших досліджень є інтеграція розроблених сервісів з Gazebo, запуск симуляції з реальними дронами. Далі буде проведено аналіз ваги феромонів ( $\alpha$ ) та ваги евристики ( $\beta$ ) на реальних прикладах, а також реалізація подвійних феромонів (привабливий та відштовхувальний) для розведення ролей між дронами у рої та розв'язання конфлікту експлуатації / дослідження. Це має значно покращити покриття, час до виявлення цілей та стійкість рою дронів. Математична формула оцінки привабливості за такої реалізації буде виглядати наступним чином [4]:

$$score_j = (\tau_j^+ + \epsilon)^\alpha * \frac{1}{1 + k\tau_j^-} * (\eta_j)^\beta, \text{ де}$$

$\tau_j^+$  – корисні феромони, які збільшують ймовірність вибору кандидату  $j$ ,

$\tau_j^-$  – штрафні феромони, які зменшують ймовірність вибору кандидату  $j$ .

#### ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ant colony optimization algorithm. URL: [https://en.wikipedia.org/wiki/Ant\\_colony\\_optimization\\_algorithms](https://en.wikipedia.org/wiki/Ant_colony_optimization_algorithms) (дата доступу 19.11.2025).
2. Introduction to Ant Colony Optimization. URL: <https://www.geeksforgeeks.org/machine-learning/introduction-to-ant-colony-optimization/> (дата доступу 19.11.2025).
3. Dorigo M, Stützle T. Ant Colony Optimization. Massachusetts Institute of Technology. 2004. P.33-38.
4. D. D. Haynes and S. M. Corns, "Algorithm for a Tabu — Ant Colony Optimizer," 2015 IEEE Congress on Evolutionary Computation (CEC), Sendai, Japan, 2015, pp. 529-535, doi: 10.1109/CEC.2015.7256935.
5. Gazebo Harmonic documentation. URL: <https://gazebo-sim.org/docs/harmonic/install/> (дата доступу 19.11.2025).