

УДК 621.391.962

Удосконалений алгоритм побудови безколізійного маршруту для багатоагентних ройових роботизованих систем

Марченко П. А., Пономаренко С. О.

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», м. Київ, Україна

E-mail: p.marchenko@kpi.ua, s.ponomarenko@kpi.ua

Зазначено, що для підвищення ефективності виконання цільових завдань застосовуються групи (рої) роботизованих засобів, що називають багатоагентними системами. Такі системи рухаються у складному середовищі з перешкодами, що потребує алгоритмізації процесу побудови безколізійних маршрутів руху роботизованих засобів для безпечного пересування у просторі. Проведено аналіз існуючих алгоритмів планування маршрутів мобільних роботизованих засобів. Показано, що для розв'язання задачі пошуку оптимального безколізійного маршруту руху групи (рою) безпілотних літальних апаратів пропонується використовувати спільну матрицю-сітку (граф) простору з перешкодами. Удосконалено алгоритм побудови маршруту багатоагентної системи роботизованих засобів для забезпечення безколізійного пересування агентів у просторі, що виконують спільні завдання у складі груп (роїв). Удосконалення проведено за рахунок додавання евристичної функції оцінювання складності переміщення між комітками. Запропоновано побудову безколізійного маршруту здійснювати за допомогою модифікованого алгоритму з видозміненою функцією евристичного оцінювання переміщення між вершинами графа, елементом якого є кубічна комітка. Показано, що основними етапами алгоритму побудови безколізійного маршруту для роя безпілотних літальних апаратів є: визначення головного елемента (лідера), побудова спільної матриці-сітки (графа) та визначення наповненості кожної комітки (вершини) в залежності від маршруту та часу, формування функції притягування/відштовхування, що визначає критерій критичного зближення елементів групи. Запропонований алгоритм побудови безколізійного маршруту для ройових багатоагентних систем дає змогу утримувати елементи групи у кластері або в складі рою в цілому під час руху у просторі з перешкодами. Подальшими дослідженнями слід вважати удосконалення математичних методів на основі використання навчання агентів з підкріпленням та моделювання процесів побудови безколізійних маршрутів для багатоагентних систем.

Ключові слова: безколізійний маршрут; багатоагентні системи; роботизовані засоби; рой; безпілотні літальні апарати; матриця-сітка; функція притягування/відштовхування; алгоритм

DOI: [10.20535/RADAP.2024.98.5-12](https://doi.org/10.20535/RADAP.2024.98.5-12)

Постановка проблеми у загальному вигляді

Сьогодні передові досягнення в мікроелектроніці, сенсорних технологіях та цифровій обчислювальній техніці визначають нові можливості розвитку робототехніки. Однією із особливостей впровадження безпілотних технологій є перехід від використання одиночних апаратів спеціалізованого призначення до застосування груп (роїв) таких апаратів, що забезпечують підвищення ефективності виконання цільових завдань [1]. З ускладненням завдань, що виконуються групами (роями) безпілотних систем, виникло поняття багатоагентної роботизованої системи, що означає групу (формацію, рій) одно- або багатотипних агентів (наземних мобільних роботів,

безпілотних летальних апаратів (БПЛА), надводних чи підводних роботизованих платформ), що виконують спільне завдання (місію). Для таких мобільних роботизованих засобів (агентів) важливим завданням є забезпечення їх безколізійного групового переміщення у складному середовищі без зіткнень між собою та перешкодами [2].

Важливим елементом під час планування та побудови маршруту для багатоагентних систем є формування кластеру роботизованих засобів, що виконують спільне завдання у складному середовищі з перешкодами. Процес формування кластеру в багатоагентних системах розповсюджений у природі, наприклад – створення зграй птахів, стад тварин, косяків риб, роїв мурах і бджіл, а також дія гравітаційних сил тіл із великою масою

щодо притягування інших тіл. Дії різних біологічних груп і фізичні явища були досліджені вченими для опису контролю груп мобільних роботизованих засобів [3–5]. Такі підходи можуть бути використанні для формування алгоритмів безколізійного планування маршрутів багатоагентних ройових роботизованих систем, зокрема, функції притягування/відштовхування об'єктів для утримування їх на безпечній відстані один від одного у кластері, не втрачаючи властивостей групової взаємодії [1, 6].

1 Аналіз останніх досліджень і публікацій

Під час групового керування постає питання передачі та синхронізації інформації стосовно загального стану багатоагентної ройової роботизованої системи та взаємодії між агентами у складі групи. Цей процес може відбуватись через обмін інформацією про положення у просторі, взаємне розташування та швидкість і напрямок руху між всіма агентами кластера, або через головного агента у групі (рою) за допомогою адаптованих протоколів взаємодії. У роботах [2, 7] для вибору головного агента у групі (рою) описані адаптовані протоколи консенсусу (Raft, Paxos), що визначають послідовність такого вибору та ситуацій, за яких потрібне переобрання такого агента.

У роботі [8] для побудови та планування маршруту групи БпЛА, простір і перешкоди у ньому поділяються на матрицю-сітку з деяким фіксованим розміром комірки, де значення елементів буде визначати зайнятість місця перешкодою або іншим агентом у просторі. Залежно від обчислювальних можливостей, складності перешкод, кількості агентів у групі (рою) та затримок під час передачі даних, гранулярність сітки та розмір видимого простору може змінюватись для більш ефективного планування маршруту та зменшення різного роду помилок.

У роботах [3, 4, 9–11] головна увага надається розробленню алгоритмів синхронізації лідера та послідовника, алгоритмів руху з сильною залежністю від головного агента групи (рою) та не розглядається побудова маршруту для рою, що утримує структуру в середовищі з перешкодами.

Для планування маршруту мобільних роботизованих засобів існує декілька варіантів алгоритмів. До таких слід віднести алгоритм Дейкстри [12], швидке дослідження випадкових дерев RТТ, RТТ* [14], пошукові алгоритми A*, D* побудови маршруту з однієї точки в іншу [6]. Зазначені алгоритми активно використовуються для розв'язання задачі побудови маршруту індивідуальних роботизованих систем у різних сферах діяльності, але мало адаптовані під проблеми групового керування рухомими об'єктами в складному середовищі з перешкодами.

Для розв'язання задачі планування та пошуку оптимального маршруту руху групи (рою) БпЛА з однієї точки в іншу запропоновано будувати спільну матрицю-сітку (граф) простору з перешкодами і плану переміщення БпЛА. Задачі локалізації БпЛА у просторі та побудови карти перешкод можуть бути розв'язані завдяки програмним продуктам з відкритою картографічною інформацією, а саме: OpenStreetMaps, алгоритмів SLAM на базі сенсорів типу Lidar та стереокамер для розширення апріорної інформації про стан середовища [14–16].

Таким чином, проведений аналіз останніх досліджень та публікацій свідчить, що на сьогодні існує низка алгоритмічних підходів для вирішення проблемної ситуації стосовно планування та побудови маршрутів для груп (роїв) роботизованих засобів, однак вони мають узагальнений характер і потребують адаптації (модифікації) методів і моделей до конкретного конструктивного виконання.

На підставі наведеного, метою й основним змістом статті є удосконалення алгоритму побудови маршруту багатоагентної системи роботизованих засобів для забезпечення безколізійного пересування агентів у просторі, що виконують спільні завдання у складі груп (роїв). Удосконалення здійснюється за рахунок додавання евристичної функції притягування/відштовхування, що характеризує складність переміщення агентів між комірками.

2 Виклад основного матеріалу

Розглянемо процес алгоритмізації побудови безколізійного маршруту для багатоагентної ройової роботизованої системи на прикладі групи (рою) БпЛА, що переміщається у обмеженому тривимірному просторі з перешкодами, для забезпечення цілісності та структури групи (рою) на всьому маршруті руху. Вважається, що задача вибору головного агента (лідера) в групі розв'язана, а передача інформації між агентами відбувається без затримок. Як критерій цілісності рою використовується функція притягування/відштовхування агентів у середовищі з відомими наперед статичними перешкодами.

2.1 Розроблення алгоритму

Побудова маршруту переміщення роботизованого засобу, зокрема БпЛА, у просторі залежить від значень матриці-сітки (графа), що містить опис статичних і активних перешкод через математичну функцію складності переміщення рою у просторі.

Процес формування матриці-сітки (графа) відбувається так: видимий простір розділяється на кубічні ділянки (комірки) розміром $d \times d \times d$ та заповнюється значеннями, що визначають наповненість комірки дискретними величинами (0) – вільна,

(1) – зайнята (в деяких випадках, можна використовувати значення у безперервному проміжку $[0, 1]$). Побудова та визначення матриці-сітки (графа) у кожній комірці відбувається з використанням інформації, що впливає на рух багатоагентної ройової роботизованої системи у просторі, а саме:

1) апіорної картографічної інформації, отриманої з карти, та доповненням непередбачуваних перешкод визначених сенсорами за методами SLAM [14, 16, 17];

2) радіусу r безпечної відстані k -го БПЛА до перешкоди (для уникнення колізій з агентами і перешкодами в середовищі) [18, 19];

3) функції притягування/відштовхування, що формулює критерій критичного зближення елементів у групі [2];

4) поточними маршрутами БПЛА, що переміщуються у складі кластера або всього рою до заданої точки, де зайнятість матриці-сітки (графа) залежить від часу.

Побудова, локалізація та орієнтування БПЛА в матриці-сітці (графі) здійснюється по Local NED системі координат [20].

Побудова безколізійного маршруту для групи (рою) роботизованих засобів потребує синхронізації станів і маршрутів усіх рухомих агентів у кластері або рою в цілому. Оскільки БПЛА рухаються у просторі – значення наповнення комірок матриці-сітки (графа) змінюється з часом. Цю інформацію можна спрогнозувати, у випадку коли є відомою швидкість руху БПЛА у просторі та визначено приблизний час прибуття t_1, t_2, \dots, t_n у кожен комірку (i, j, m) матриці-сітки (графа) розміром d . Водночас, слід дотримуватись між агентами безпечної відстані, що позначена зоною з радіусом r . Приклад матриці-сітки (графу) з наповненням комірок, що залежить від маршруту агента на визначеній висоті та часу переміщення, наведений на Рис. 1.

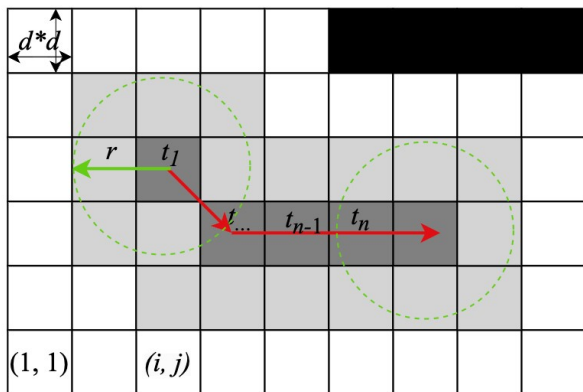


Рис. 1. Наповненість двовимірної матриці-сітки (графа) залежно від маршруту агента та часу його руху

На Рис. 1 позначені комірки: чорним кольором – статична перешкода, сірим – позиція агента uav , світло-сірий – зона безпечної дистанції.

Функція стану наповненості комірки (*Occupancy*) для тривимірної матриці-сітки (графа) залежить від часу руху агента (uav) у складі кластера або всього рою за поточним маршрутом (Рис. 1). Зазначену функцію доцільно визначати за таким виразом:

$$Occupancy(gridmap, \langle i, j, m \rangle, route, uav) = \begin{cases} 1 & \text{if } route \not\# uav, route \ni gridmap(i, j, m), \\ t_{n-1} - \delta < t_{est} < t_n + \delta; \\ 0 & \text{otherwise,} \end{cases} \quad (1)$$

де $gridmap$ – матриця-сітка з усіма видимими позиціями групи (рою) БПЛА;

$route$ – маршрут, що визначається набором позицій i, j, m ;

uav – агент кластеру або рою в цілому;

i, j, m – індекси комірки матриці-сітки в Local NED системі координат [20];

t_{est} – час, на момент якого обчислюється зайнятість комірки;

t_{n-1}, t_n – моменти часу, коли uav займає позицію в комірці з індексом i, j, m ;

δ – проміжок часу, що визначає безпечний інтервал використання маршруту різними агентами uav .

Утримання БПЛА в кластері або всьому рою на контрольованій відстані один від одного відбувається завдяки видозміненій евристичній функції потенціалу (*Potential*) притягування/відштовхування, що визначає її значення в комірці матриці-сітки (графа) та формулює критерій критичного зближення агентів [5]:

$$Potential(y) = -y \left(\frac{a}{||y||^2} - \frac{b}{||y||^4} \right) - \min Potential(y), \quad (2)$$

де $y \in N^3$ – відстань від центру БПЛА у видимому просторі;

$a, b \in R$ – гіпер-параметри, що задають роботу відстань і потенціал поля та визначаються емпірично.

Графік функції притягування/відштовхування (*Attraction/repulsion*) залежно від відстані y (*Distance*), яку доцільно використовувати під час побудови маршруту наведено на Рис. 2 [1].

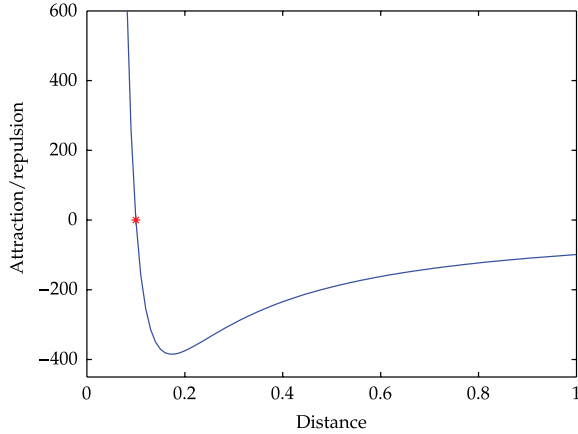


Рис. 2. Функція притягання/відштовхування залежно від відстані y , м

При побудові маршруту БпЛА у видимому просторі комірки матриці-сітки (графа), що зайняті перешкодами або іншими БпЛА, виключаються з простору пошуку маршруту для уникнення потенційних колізій (зіткнень). Визначивши вільний простір і складність переміщення агентів через значення в матриці-сітці (Рис. 1), побудуємо маршрут руху за удосконаленим алгоритмом A^* [6], модифікованим за рахунок додавання евристичної функції потенціалу (*Potential*) притягання/відштовхування, що визначає ройовий фактор складності переміщення агента у просторі. Тому функцію, що використовується в машинному навчанні для балансування точності і відгуку, доцільно визначити за виразом:

$$fScore(cell) = H(cell) + D(cell) + Potential(cell), \quad (3)$$

де $cell$ – наступна комірка на маршруті руху, що визначається алгоритмом;

H – евристична функція оцінювання відстані від наступної комірки до кінця маршруту;

D – функція евклідової відстані між поточною та наступною комірками;

Potential – евристична функція притягання/відштовхування, що характеризує складність переміщення агентів між комірками.

Реалізація алгоритму A^* побудови безколізійного маршруту для переміщення БпЛА у просторі, де пріоритетність черги $minHeapSet$ відбувається по визначеному результату функції $fScore$:

```
function reconstruct_path(cameFrom, current)
    total_path := {current}
    while current in cameFrom.Keys:
        current := cameFrom[current]
        total_path.prepend(current)
    return total_path
function uav_find_path(t_uav, uav_map, start, goal, H)
    minHeapSet := {start}
    cameFrom := []
```

```
gScore := Infinity
gScore[start] := 0
fScore := Infinity
fScore[start] := H(start) + uav_map(start).pot
while minHeapSet is not empty
    current := minHeapSet.get()
    if current = goal
        return reconstruct_path(cameFrom, current)
    minHeapSet.Remove(current)
    for each neighbor of current
        tentative_gScore := gScore[current] +
            D(current, neighbor) +
            uav_map(neighbor).pot
        if tentative_gScore < gScore[neighbor]
            cameFrom[neighbor] := current
            # евристична функція оцінювання
            # складності досягнення позиції
            gScore[neighbor] := tentative_gScore
            fScore[neighbor] := tentative_gScore +
                H(neighbor)
            if neighbor not in minHeapSet
                minHeapSet.add(neighbor)
return emptySet
```

Навігаційна система кожного БпЛА забезпечує визначення потрібних навігаційних параметрів (координат x , y , z , компонентів швидкості V_x , V_y , V_z , кути крену, тангажу та ризкання) на кожному етапі переміщення агента по заданому маршруту [11, 19, 21–23]. Загальна схема удосконаленого алгоритму A^* побудови безколізійного маршруту багатоагентної ройової роботизованої системи, що реалізує вирази (1)–(3) наведена на Рис. 3.

Основними етапами розробленого алгоритму (Рис. 3) побудови безколізійного маршруту для групи (роя) БпЛА у середовищі з перешкодами є визначення головного елемента (лідера) за відомим протоколом консенсусу (Raft), побудова спільної матриці-сітки (графа) та визначення наповненості кожної комірки залежно від маршруту та часу руху, формування евристичної функції притягання/відштовхування, що визначає критерій критичного зближення агентів у кластері (рою) та оцінює складність переміщення до наступної комірки.

До евристичної функції оцінювання переміщення між вершинами графу додається ройовий фактор, що визначає потенціал притягання/відштовхування роботизованого засобу, що залежить від відстані до іншого агента або перешкоди. Побудову маршруту та формування виконання команди на переміщення пропонується здійснювати безпосередньо на роботизованому засобі, що дає змогу перейти у автономний режим за умови втрати зв'язку з лідером. Формування команди керування БпЛА відбувається за допомогою визначених компонент швидкості та інтерфейсом інерціальної навігаційної системи, що визначається типом безпілотного апарата.

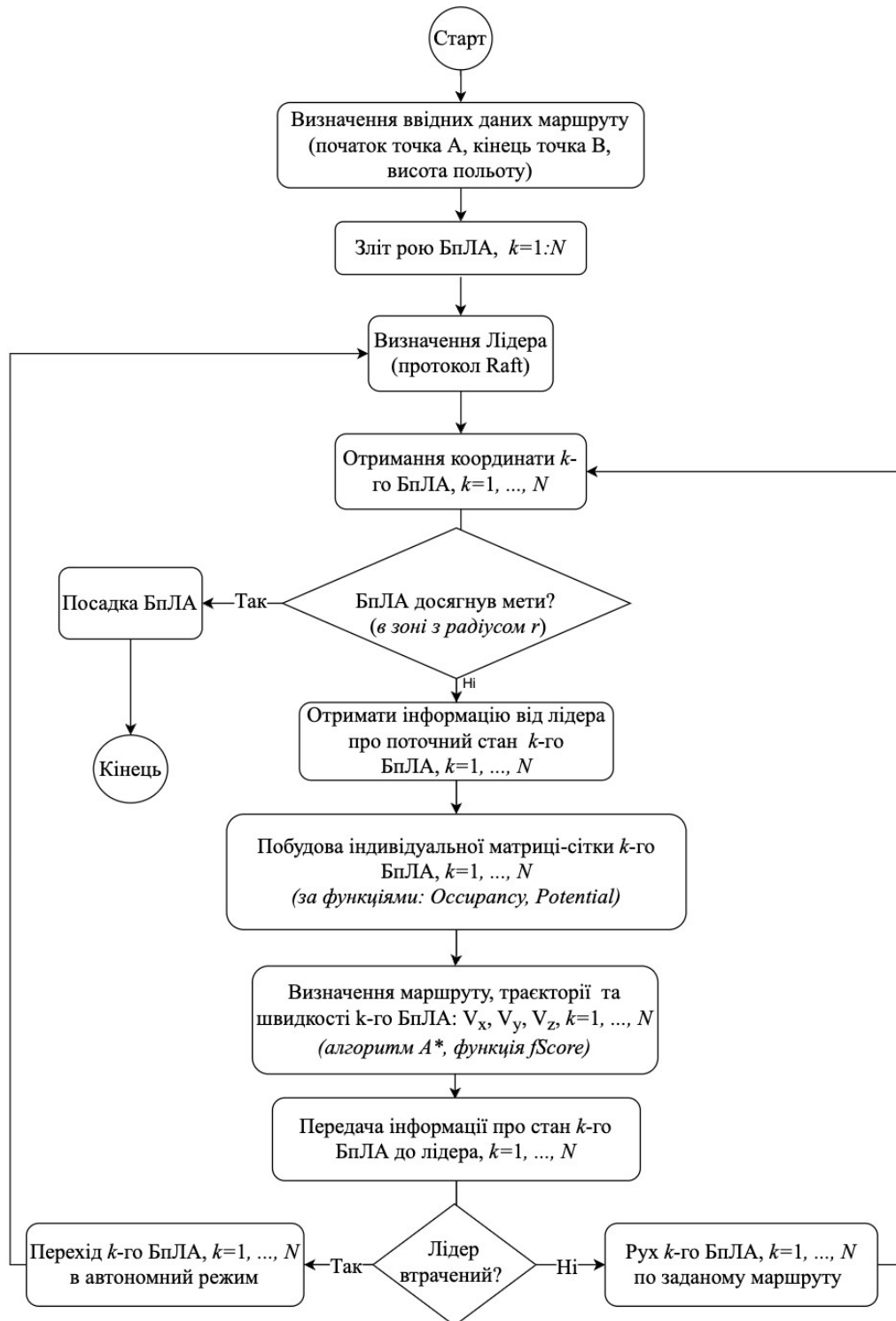


Рис. 3. Загальна схема алгоритму побудови безколізійного маршруту

2.2 Моделювання в середовищі мови програмування Python

Для перевірки працездатності запропонованого алгоритму було проведено моделювання у середовищі мови програмування Python та бібліотеки matplotlib на платформі процесора ARM Cortex-A72 (1.8 ГГц). Візуалізація моделює процес побудо-

ви безколізійного маршруту та переміщення групи з трьох БПЛА у просторі розміром $10 \times 15 \times 5$ м з видимими перешкодами. Роздільна здатність матриці сітки становить $d = 1$ м. Параметри для функції *Potential* (2) становлять $a = 10$ та $b = 50$ відповідно. Під час розрахунків було встановлено, що побудова безколізійного маршруту досягається за оптимального значення параметру $\delta(1) = 1$ с. Згладжування

шляху виконано за допомогою методу кубічного сплайну з гіперпараметром $N_{points} = 5$. Результати моделювання за розробленим алгоритмом та його 3D візуалізація наведені на Рис. 4.

За результатами роботи алгоритму встановлено, що загальний час T_{total} побудови маршруту не перевищує 30 мс, а максимальне значення похибки визначення часу побудови маршруту ΔT_{total} становить $-4,9\%$.

Особливістю та перевагою алгоритму побудови безколізійного маршруту багатоагентної ройової роботизованої системи (Рис. 3), розробленого для спільного руху БПЛА у складі групи (рою) є використання евристичної функції притягування/відштовхування та адаптованого алгоритму пошуку, що забезпечує їх рух без взаємного зіштовхування у середовищі зі статичними перешкодами.

Для нівелювання обмежених обчислювальних можливостей і пришвидшення розрахунків доцільно надати такі пропозиції:

збільшення розміру комірки матриці-сітки (графа) та зменшення кількості вершин у графі, що призведе до зменшення кількості ітерацій при побудові маршруту;

використання критерію “обмеженого розслаблення” в алгоритмі A* гарантує розв’язання задачі пошуку маршруту на $(1 + \varepsilon)$ не гірше ніж оптимальний, що призведе до зменшення точності алгоритму. При цьому ε – компромісне значення між швид-

кістю алгоритму та його якістю, що визначається емпірично.

2.3 Перспективи подальшого розвитку дослідження

Одним з напрямів подальших досліджень слід вважати удосконалення методів на основі використання навчання агентів з підкріпленням та моделювання процесів побудови безколізійних маршрутів для багатоагентних ройових роботизованих систем.

Висновки

1. Отже в основу запропонованого алгоритму побудови безколізійного маршруту багатоагентних ройових роботизованих систем, зокрема БПЛА, що виконують спільні завдання закладено підхід побудови матриці-сітки (графа), що розділяє простір на комірки. Наповненість кожної комірки визначається залежно від маршруту БПЛА та часу переміщення у просторі. Критичне зближення агентів кластеру або рою визначається функцією притягування/відштовхування, а складність переміщення до наступної комірки – евристичною функцією оцінювання переміщення роботизованих засобів між вершинами графа.

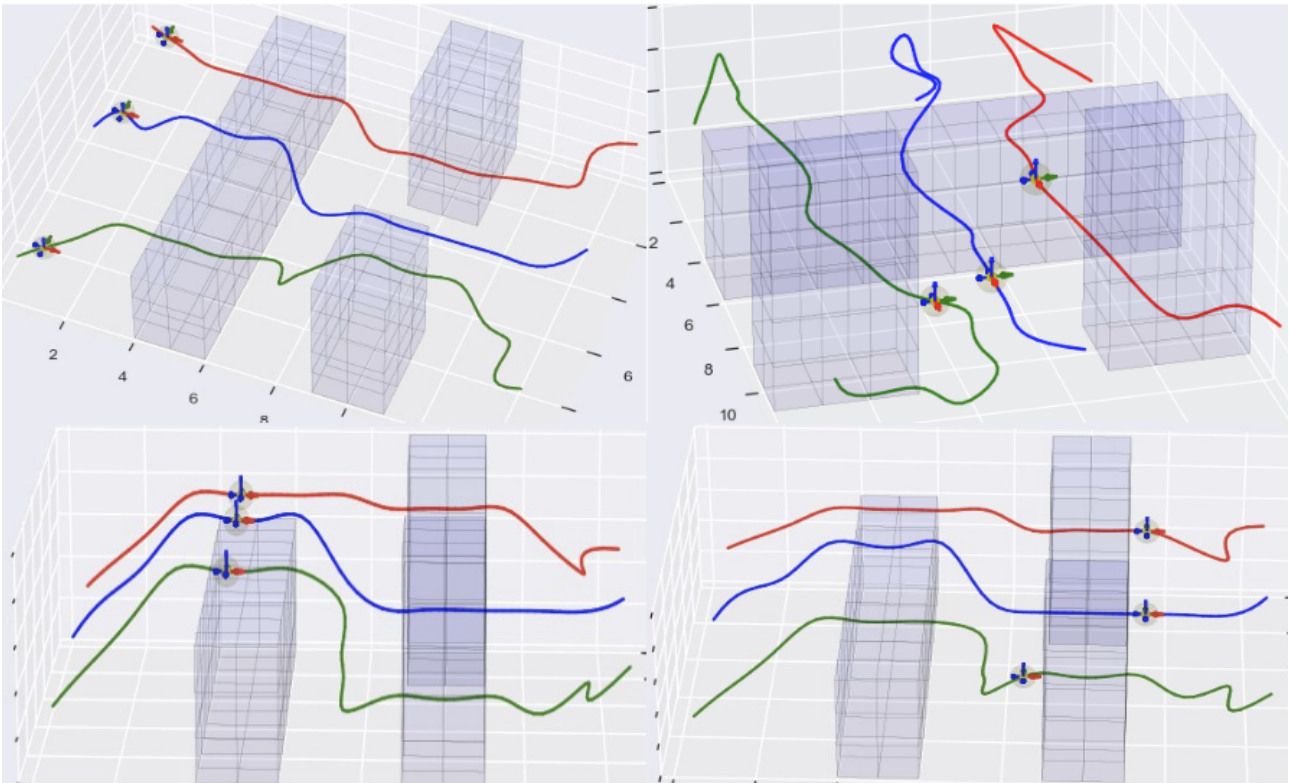


Рис. 4. Результати моделювання та його 3D візуалізація

2. Запропонований алгоритм побудови безколізійного маршруту для багатоагентних ройових роботизованих систем, реалізований у вигляді програмного продукту, дає змогу забезпечити цілісність роя БПЛА і утримання його агентів у складі єдиного кластеру, безпечно пересування БПЛА (без зіткнень між ними) у просторі з статичними (нерухомими) перешкодами при виконанні спільних завдань. Проведене моделювання підтвердило адекватність обраного підходу до алгоритмізації процесу побудови безколізійного руху агентів у складі роя.

References

- [1] Shi H., Xie G. (2011). Collective Dynamics of Swarms with a New Attraction/Repulsion Function. *Mathematical Problems in Engineering*, Vol. 2011, Article ID 735248, 13 p. DOI: 10.1155/2011/735248.
- [2] Tariverdi A., Torresen J. (2023). Rafting Towards Consensus: Formation Control of Distributed Dynamical Systems. *arXiv.org*. DOI: 10.48550/arXiv.2308.10097.
- [3] Lawton J. R. T., Beard R. W., Young B. J. (2003). A decentralized approach to formation maneuvers. *IEEE Trans. Robot. Autom.*, Vol. 19, Iss. 6, pp. 933–941. DOI: 10.1109/TRA.2003.819598.
- [4] Kojima, M., Nakano, H., Miyauchi, A. (2013). An Artificial Bee Colony algorithm for solving dynamic optimization problems. *2013 IEEE Congress on Evolutionary Computation (CEC)*. DOI: 10.1109/cec.2013.6557856.
- [5] Yamanaka Y., Yoshida K. (2021). Simple gravitational particle swarm algorithm for multimodal optimization problems. *PLoS ONE*, Vol. 16(3): e0248470. DOI: 10.1371/journal.pone.0248470.
- [6] Duchon F., Babinec A., Kajan M., Beno P., Florek M., Fico T., Jurišica L. (2014). Path Planning with Modified A* Algorithm for a Mobile Robot. *Procedia Eng.*, Vol. 96, pp. 59–69. DOI: 10.1016/j.proeng.2014.12.098.
- [7] Arnaud Favier. (2022). Eventual Leader Elections in Dynamic Networks. Distributed, Parallel, and Cluster Computing [cs.DC]. *HAL open science*, Sorbonne Université, NNT:2022SORUS059, tel-03624018v2.
- [8] Sabziev E. (2021). A control algorithm for joint flight of a group of drones. *Scientific Journal of Silesian University of Technology*, Series Transport, Vol. 110, pp. 157–167. DOI:10.20858/sjsutst.2021.110.13.
- [9] Jadbabaie A., Lin J., Morse A. S. (2003). Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Trans. Autom. Control*, Vol. 48, Iss. 6, pp. 998–1001. DOI: 10.1109/TAC.2003.812781.
- [10] Pan H., Zhou R., Zhang S. (2023). Research on Formation Control Method of UAV Group. *IEEE International Conference on Control, Electronics and Computer Technology (ICCECT)*, pp. 157–162. DOI: 10.1109/ICCECT57938.2023.10141302.
- [11] Zakhari F. M., Ponomarenko S. A. (2017). Unmanned Aerial Vehicle Integrated Navigation Complex. *Electronics and control systems*, Vol. 3, Iss. 53, pp. 75–83. DOI:10.18372/1990-5548.53.12146.
- [12] Wang, H.; Yu, Y.; Yuan, Q. (2011). Application of Dijkstra algorithm in robot path-planning. *In Proceedings of the 2011 2nd International Conference on Mechanic Automation and Control Engineering*, pp. 1067–1069. DOI: 10.1109/MACE.2011.5987118.
- [13] Karaman S., Walter M., Perez A., Frazzoli E., Teller S. (2011). Anytime motion planning using the RRT*. *2011 IEEE International Conference on Robotics and Automation*, pp. 1478–1483. DOI: 10.1109/ICRA.2011.5980479.
- [14] Frosi, M., Gobbi, V., Matteucci, M. (2023). OSM-SLAM: Aiding SLAM with OpenStreetMaps priors. *Frontiers in Robotics and AI*, Vol. 10. DOI:10.3389/frobt.2023.1064934.
- [15] Huang, L. (2021). Review on LiDAR-based SLAM Techniques. *2021 International Conference on Signal Processing and Machine Learning (CONF-SPML)*, pp. 163–168. DOI: 10.1109/CONF-SPML54095.2021.00040.
- [16] Engel, J., Stuckler, J., Cremers, D. (2015). Large-scale direct SLAM with stereo cameras. *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. DOI: 10.1109/IROS.2015.7353631.
- [17] Meyer-Delius, D., Beinhofer, M., Burgard, W. (2021). Occupancy Grid Models for Robot Mapping in Changing Environments. *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 26, Iss. 1, pp. 2024–2030. DOI: 10.1609/AAAI.V26I1.8377.
- [18] Rogers, A., Eshaghi, K., Nejat, G., Benhabib, B. (2023). Occupancy Grid Mapping via Resource-Constrained Robotic Swarms: A Collaborative Exploration Strategy. *Robotics*, Vol. 12, Iss. 3, 70. DOI:10.3390/robotics12030070.
- [19] Amanatides, J., Woo, A. (1987). A Fast Voxel Traversal Algorithm for Ray Tracing. *Eurographics 1987-Technical Papers*, pp. 3–10. DOI: 10.2312/egtp.19871000.
- [20] Cai, G., Chen, B. M., Lee, T. H. (2011). Coordinate Systems and Transformations. In: *Unmanned Rotorcraft System*, Springer, pp. 23–34. DOI: 10.1007/978-0-85729-635-1_2.
- [21] Ren H., Chen S., Yang L., Zhao Y. (2020). Optimal Path Planning and Speed Control Integration Strategy for UGVs in Static and Dynamic Environments. *IEEE Transactions on Vehicular Technology*, Vol. 69, Iss. 10, pp. 10619–10629. DOI: 10.1109/tvt.2020.3015582.
- [22] Ahmed A., Ouda A., Kemel A., Elhalwagy Y. Z. (2015). Design and Analysis of Quadcopter Classical Controller. *International Conference on Aerospace Sciences & Aviation Technology, ASAT-16*. DOI: 10.21608/asat.2015.23032.
- [23] Abualigah L., Izci D., Ekinici S., Zitar R. A. (2024). Optimizing Aircraft Pitch Control Systems: A Novel Approach Integrating Artificial Rabbits Optimizer with PID-F Controller. *International Journal of Robotics and Control Systems*, Vol. 4, No. 1, pp. 354–364. DOI: 10.31763/ijrcs.v4i1.1347.

Improved Algorithm for Constructing Collision-Free Route for Multi-Agent Swarm Robotic Systems

Marchenko P. A., Ponomarenko S. O.

Formulation of the problem in general.

It is noted that to increase the efficiency of performing targeted tasks, groups (swarms) of robots, called multi-agent systems, are used. Such systems move in a complex environment with obstacles, which requires algorithmization of the process of building collision-free routes for the movement of robotic vehicles for safe movement in space. An analysis of existing algorithms for planning routes for mobile robotic vehicles is carried out. It is shown that to solve the problem of finding the optimal collision-free route for a group (swarm) of unmanned aerial vehicles, it is proposed to use a joint matrix-grid (graph) of the space with obstacles and the movement plan.

Analysis of recent researches and publications.

The Dijkstra algorithm, fast random tree search, and search algorithms for building a route from one point to another are used to plan the route of mobile robotic vehicles. However, they solve the problems of route planning for individual robotic systems in various fields of activity and are not well adapted to the problems of group control of moving objects in a complex environment with obstacles.

Presenting the main material.

The algorithm for constructing a route for a multi-agent system of robotic vehicles has been improved to ensure collision-free movement of agents in space performing common tasks as part of groups (swarms). The improvement is made by adding a heuristic function for

estimating the complexity of movement between cells. It is proposed to build a collision-free route using a modified algorithm with a modified function of heuristic evaluation of movement between the vertices of a graph, which is an element of a cubic cell. It is shown that the main stages of the algorithm for constructing a collision-free route for a swarm of unmanned aerial vehicles are: determining the main element (leader), constructing a common grid matrix (graph) and determining the occupancy of each cell (vertex) depending on the route and time, forming an attraction/repulsion function that determines the criterion for critical convergence of group elements.

Conclusion.

The algorithm for constructing a collision-free route for swarming multiagent systems allows us to keep the group elements in a cluster or as part of the swarm as a whole while moving in a space with obstacles. The proposed algorithm is based on the approach of constructing a grid matrix (graph) that divides the space into cells. The occupancy of each cell is determined depending on the UAV's route and travel time. The criterion for critical convergence of group elements is determined by the attraction/repulsion function, and the difficulty of moving to the next cell is determined by a heuristic function for estimating the movement between the vertices of the graph.

The perspectives of future researches.

Further research should include the improvement of mathematical methods based on the use of reinforcement learning of agents and modeling the processes of building collision-free routes for multi-agent systems.

Keywords: collision-free route; multi-agent systems; robotic vehicles; swarm; unmanned aerial vehicles; grid matrix; attraction/repulsion function; algorithm