

THUẬT TOÁN LẬP QUỸ ĐẠO DI CHUYỂN NHANH VÀ ĐƠN GIẢN CHO THIẾT BỊ TỰ HÀNH

A FAST AND SIMPLE TRAJECTORY PLANNING ALGORITHM FOR AUTOMOTIVE DEVICES

Nguyễn Ngọc Tuấn^{1*}, Tăng Thanh Lâm¹,
Lại Tiến Đệ¹, Trần Xuân Tình²

TÓM TẮT

Xác định quỹ đạo di chuyển từ vị trí A đến B trong môi trường động, đảm bảo tránh được vật cản là một nhiệm vụ rất phức tạp đối với các thiết bị tự hành. Bài báo này đề xuất một thuật toán lập quỹ đạo di chuyển (trajectory planning) tốc độ cao, yêu cầu cài đặt đơn giản, đảm bảo an toàn cho thiết bị tự hành có phần cứng không cao.

Từ khóa: Lập quỹ đạo di chuyển, tìm đường ngắn nhất, xe tự lái, thiết bị tự hành.

ABSTRACT

Determining the trajectory moving from position A to B, while avoiding obstacles as well as changing the environment is a simple task for humans but very complex for autonomous devices. Typically, autonomous devices will use sensors to build maps of the surrounding environment, through which to determine the trajectory of movement as well as appropriate control action to reach the desired destination. This article proposes a high-speed trajectory planning algorithm that requires simple settings to ensure safety for self-propelled equipment with low hardware requirements.

Keywords: Motion Planning, shortest path searching, self-driving car, autonomous equipment.

¹Học viện Kỹ thuật Quân sự

²Học viện Phòng không Không quân

*Email: ngoctuanhvhn@gmail.com

Ngày nhận bài: 29/10/2020

Ngày nhận bài sửa sau phản biện: 10/12/2020

Ngày chấp nhận đăng: 26/02/2021

1. GIỚI THIỆU

Nhưng năm gần đây, các thiết bị tự hành đang dần được đưa vào ứng dụng trong thực tiễn nhiều hơn thay vì chỉ hoạt động trong các môi trường chuyên biệt như các dây chuyền sản xuất công nghiệp. Từ đó đặt ra các yêu cầu đối với các thiết bị tự hành có khả năng làm việc linh hoạt trước sự thay đổi của môi trường xung quanh. Do đó, giải quyết bài toán lập quỹ đạo di chuyển an toàn, tránh vật cản tĩnh và động đối với các thiết bị tự hành có vai trò đặc biệt quan trọng. Các thuật toán này thường được gọi là thuật toán lập kế hoạch di chuyển (path planning) hay thuật toán tìm đường và là vấn đề đã được nghiên cứu lâu dưới dạng bài toán miền liên thông, bài toán tìm đường ngắn nhất trên đồ thị... Nhiều thuật toán đã giải quyết trọn vẹn bài

toán với kết quả là một lộ trình tối ưu với nhiều phương pháp tiếp cận khác nhau. Trong thực tế, môi trường xung quanh không phải lúc nào cũng được biết một cách đầy đủ hay có thể được mô tả một cách chính xác trên đồ thị. Ngoài ra, đối với các thiết bị tự hành, vấn đề về tính toán thời gian thực trước sự thay đổi của môi trường cũng là một yêu cầu bắt buộc. Bài báo đề xuất một thuật toán lập quỹ đạo nhanh, đơn giản, độ phức tạp thấp nhằm giải quyết các vấn đề trên.

2. TỔNG QUAN VỀ CÁC GIẢI THUẬT LẬP QUỸ ĐẠO DI CHUYỂN ĐÃ ĐƯỢC NGHIÊN CỨU

Lập quỹ đạo di chuyển chỉ có thể được áp dụng khi bản đồ về môi trường được biết trước một cách đầy đủ hoặc một phần. Do đó, robot phải có khả năng định vị và lập bản đồ thời gian thực mới có thể lập áp dụng các thuật toán xây dựng quỹ đạo di chuyển tối ưu [1, 2, 3]. Có rất nhiều cách phân loại các thuật toán lập quỹ đạo di chuyển, tuy nhiên, ta có thể chia chúng làm hai loại chính đó là lập quỹ đạo di chuyển trên cơ sở hình học không gian (geometric-based planner) và lập quỹ đạo di chuyển trên cơ sở điều khiển (control-based planner) [4].

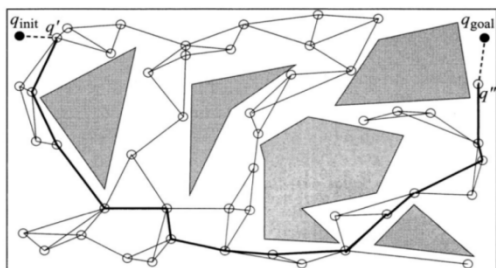
2.1. Phương pháp lập quỹ đạo di chuyển trên cơ sở hình học không gian (geometric - based planner)

Các quỹ đạo di chuyển được lập dựa trên phương pháp này chỉ quan tâm đến các ràng buộc về hình học. Người ta cho rằng bất kỳ con đường khả thi nào về hình học cũng có thể chuyển đổi về một quỹ đạo khả thi về động học [5]. Kết quả của phương pháp này là một con đường không va chạm giữa điểm ban đầu và mục tiêu cuối cùng của thiết bị với số lượng điểm tham chiếu tối thiểu, là một mô ta hình học chuyển động của thiết bị, tọa độ (x, y, z) của các điểm nổi trên lộ trình. Lập quỹ đạo trên cơ sở hình học có thể chia làm hai nhóm: *Muli-query planners* và *Single-query planners*, trong đó:

Muli-query planners: Nhiều lộ trình được tính toán đồng thời với thuật toán dừng lại khi một lộ trình đến đích, tương tự như thuật toán tìm kiếm theo chiều rộng (BFS). Thuật toán tiêu biểu theo phương pháp này là: Probability Roadmap Method (PRM).

Single-query planners: Các hướng di chuyển tiềm năng được thêm vào cây quyết định và được duyệt lần lượt theo ưu tiên nào đó và kết thúc khi thỏa mãn các điều kiện ràng

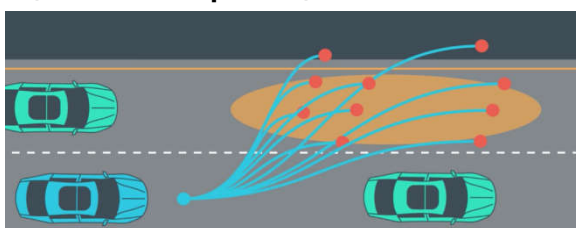
buộc. Khi tất cả các quỹ đạo chuyển động được xây dựng, thuật toán sẽ đánh giá các quỹ đạo đến được đích và đưa ra kết quả là quỹ đạo có chi phí di chuyển thấp nhất.



Hình 1. Ví dụ kết quả lập quỹ đạo di chuyển của một thuật toán lập quỹ đạo di chuyển trên cơ sở hình học

Có thể thấy, lộ trình mà các thuật toán của phương pháp này đưa ra là các đường gấp khúc nối từ điểm đầu đến điểm cuối của lộ trình và cần chuyển đổi sang quỹ đạo di chuyển mượt hơn cho phù hợp với dạng di chuyển của thiết bị. Tuy nhiên, quá trình chuyển đổi này làm tăng độ phức tạp tính toán của thuật toán và có thể làm mất đi sự tối ưu ban đầu của lộ trình.

2.2. Phương pháp lập quỹ đạo di chuyển trên cơ sở điều khiển (control-based planner)

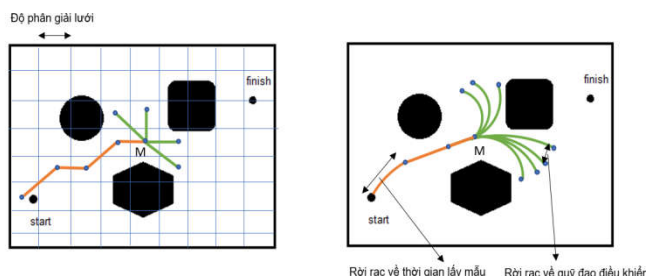


Hình 2. Quỹ đạo di chuyển được đưa ra theo phương pháp lập quỹ đạo trên cơ sở điều khiển

Phương pháp lập quỹ đạo di chuyển được sử dụng khi các điều kiện về động lực học hay điều khiển được xét đến ngay trong quá trình xây dựng quỹ đạo di chuyển. Trong khi lập quỹ đạo di chuyển dựa trên cơ sở hình học là một chuỗi các điểm biểu diễn cấu hình hình học hay tọa độ các điểm nối trong không gian của lộ trình thì phương pháp lập quỹ đạo trên cơ sở điều khiển đưa ra kết quả là một chuỗi các quỹ đạo di chuyển con biểu diễn quỹ đạo thực tế lộ trình di chuyển cần thiết.

Có thể thấy, so với phương pháp lập quỹ đạo di chuyển trên cơ sở hình học, phương pháp này cho kết quả lộ trình trực quan, chính xác và có thể sử dụng được ngay để đưa ra quyết định điều khiển. Mặc dù vậy, khi tính toán trên thực tế, các thuật toán dựa trên cơ sở hình học thường sử dụng các kỹ thuật rời rạc không gian thành dạng lưới hoặc các miền liên thông từ đó giảm được đáng kể chi phí tính toán mà vẫn đảm bảo độ tối ưu cao [6].

Trong khi đó, các thuật toán dựa trên cơ sở điều khiển mặc dù có thể rời rạc hóa bài toán bằng cách rời rạc hóa quỹ đạo điều khiển kết hợp với rời rạc hóa thời gian tính quỹ đạo con nhưng độ phức tạp tính toán vẫn rất lớn vì có quá nhiều quỹ đạo phù hợp cần xét đến.



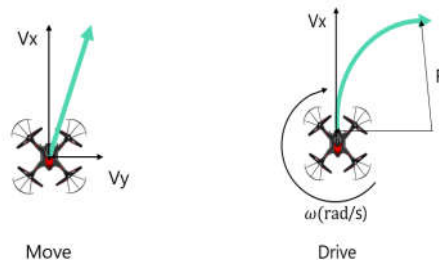
Hình 3. So sánh các kỹ thuật rời rạc hóa trên cơ sở hình học (trái) và trên cơ sở điều khiển (phải)

Việc tính toán các quỹ đạo con và kiểm tra quỹ đạo có xảy ra va chạm hay không cũng là một vấn đề khó khăn làm tăng độ phức tạp đối với phương pháp xây dựng quỹ đạo di chuyển dựa trên cơ sở điều khiển.

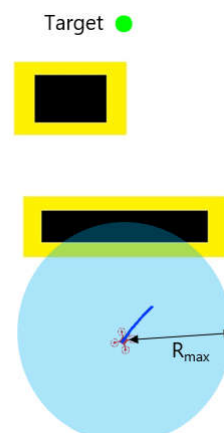
3. ĐỀ XUẤT THUẬT TOÁN LẬP QUỸ ĐẠO DI CHUYỂN NHANH, ĐƠN GIẢN CHO THIẾT BỊ TỰ HÀNH

3.1. Cơ sở của thuật toán

Thuật toán lập quỹ đạo di chuyển do nhóm tác giả đề xuất trong bài báo này thuộc phương pháp lập quỹ đạo di chuyển trên cơ sở điều khiển. Để giảm thiểu số quỹ đạo cần tính toán, nhóm tác giả đề xuất xây dựng thuật toán trên cơ sở kết hợp ý tưởng của hai nhóm *Multi-query planners* và *Single-query planners*. Ngoài ra nhóm tác giả cũng đưa ra phương pháp tính nhanh quỹ đạo con và kiểm tra va chạm cho quỹ đạo con nhanh làm giảm độ phức tạp của thuật toán.



Hình 4. Chế độ điều khiển move (trái) và chế độ điều khiển drive (phải) của drone



Hình 5. Bài toán lập quỹ đạo di chuyển cho drone đến đích (màu đen minh họa vật cản, màu vàng là vùng nguy hiểm)

Để thuận tiện cho việc trình bày thuật toán, ta xét một ví dụ cụ thể đối với bài toán điều khiển drone đi đến mục

tiêu, tránh vật cản trong không gian 2D. Giả sử drone được trang bị các cảm biến cần thiết và có thể định vị cũng như lập bản đồ môi trường xung quanh trong một bán kính giới hạn R_{max} . Ta chỉ xét drone đối với trường hợp điều khiển theo chế độ lái (drive) - với tốc độ tiến V_x (m/s) và tốc độ góc quay ω (rad/s). Trong trường hợp chế độ điều khiển move (di chuyển tịnh tiến) hoặc kết hợp cả hai chế độ ta có thể tiến hành tương tự.

3.2. Một số thuật toán phụ cần thiết

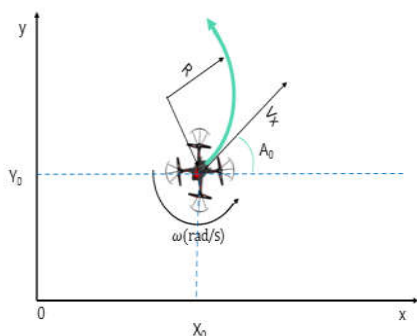
Thuật toán 1: Tìm quỹ đạo di chuyển và trạng thái của drone ứng với vận tốc tiến V_x (m/s) và tốc độ vận tốc quay ω (rad/s).

$$R = (2\pi/\omega) * V_x / 2\pi = V_x / \omega$$

Có thể thấy quỹ đạo R của drone phụ thuộc vào tỉ lệ V_x/ω và ngược lại. Ví thể, khi nhắc đến quỹ đạo, ta có thể coi như nói đến tỉ số V_x/ω (V_x, ω có dấu). Giả sử vị trí hiện tại của drone là (X_0, Y_0) , hướng tiến của drone hợp với trục Ox 1 góc A_0 (rad). Trạng thái mới của drone sau khoảng thời gian dt là:

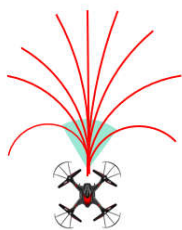
$$A_1 = A_0 + \omega * dt; X_1 = X_0 + R * (\cos A_0 - \cos A_1);$$

$$Y_1 = Y_0 + R * (\sin A_0 + \sin A_1)$$



Hình 6. Tính quỹ đạo chuyển động R theo lệnh điều khiển V_x, ω

Thuật toán 2: Đề xuất các quỹ đạo di chuyển:



Hình 7. Ví dụ các quỹ đạo di chuyển được xét đến

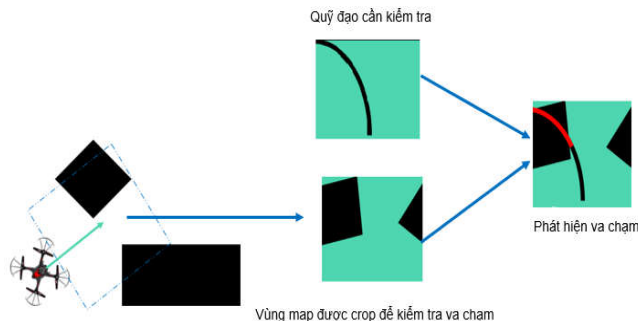
Để rời rạc bài toán, quỹ đạo di chuyển của drone được rời rạc hóa. Trong quá trình kiểm tra các quỹ đạo con thỏa mãn, ta chỉ xem xét các quỹ đạo con này.

Các quỹ đạo rời rạc đã được tính toán trước vì chúng là không đổi nên giảm thiểu độ phức tạp thuật toán khi chạy.

Thuật toán 3: Kiểm tra va chạm nhanh sử dụng phương pháp xử lý ảnh

Các quỹ đạo con được tính toán trước như ở thuật toán 3, đồng thời, mỗi thuật toán con tương ứng với một mask

dùng để kiểm tra va chạm, các mask này cũng được tạo ra và lưu trữ trước. Drone được định vị trên bản đồ (environment map) về vị trí và góc xoay, vùng bản đồ phía trước drone được crop ra để kiểm tra va chạm. Vùng map đã crop được resize đúng kích thước của mask quỹ đạo, khi đó để kiểm tra 1 quỹ đạo có xảy ra va chạm hay không, ta chỉ cần dùng hàm XOR 2 ảnh, nếu có pixel nào trùng giữa 2 ảnh, ta sẽ biết quỹ đạo đó là va chạm.



Hình 8. Ví dụ kiểm tra va chạm cho quỹ đạo con với một trạng thái của drone

3.3. Thuật toán lập quỹ đạo di chuyển

Tại 1 trạng thái bất kỳ của drone, để tìm quỹ đạo di chuyển đến mục tiêu, tư tưởng của thuật toán như sau:

1. Tại mỗi thời điểm, để tìm đường đi tốt nhất, drone sẽ thực hiện tính toán các quỹ đạo được tạo bởi M quỹ đạo con (tính trước M bước). Nếu mục tiêu ở quá xa vị trí hiện tại của drone, các quỹ đạo đưa ra của thuật toán chỉ là các quỹ đạo có khả năng tốt hướng đến mục tiêu chưa hoàn thành quỹ đạo hoàn chỉnh đến mục tiêu. Điều này phù hợp hơn trong thực tế vì phạm vi cảm biến để xây dựng bản đồ môi trường là hạn chế, nên mục tiêu có thể nằm ngoài phạm vi này. Hơn nữa, môi trường ở đây là môi trường động, khi tính toán trước quá nhiều bước, kết quả của các bước cuối cùng (tương lai xa) sẽ thiếu chính xác hay không có nhiều ý nghĩa nữa.

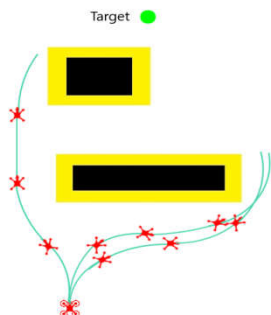
2. Tại bước đầu tiên, từ trạng thái hiện tại của drone, ta chọn N quỹ đạo con tốt nhất theo một số tiêu chí nào đó (các quỹ đạo này không xảy ra va chạm). Từ N quỹ đạo con đó, ta tính toán N trạng thái tiếp theo của drone. Trong K-1 bước tiếp theo, ta thực hiện tương tự như thế với trạng thái đã tính toán của bước trước, ta tiếp tục chọn N quỹ đạo con tốt nhất.

3. Tại M-K bước cuối cùng, với mỗi trạng thái được tính toán ở bước trước, ta chỉ chọn 1 quỹ đạo con tốt nhất.

4. Như vậy ta có tất cả N^K quỹ đạo, mỗi quỹ đạo được xây dựng từ M quỹ đạo con. Ta sẽ đánh giá các quỹ đạo đó theo một số tiêu chí mong muốn như tổng quãng đường ngắn nhất, lộ trình mượt nhất, trạng thái cuối cùng của drone để đến được đích là thuận tiện nhất (gần đích nhất, hướng tiến của drone và hướng tiến về đích tạo thành góc nhỏ nhất,...). Cuối cùng, Drone sẽ chọn quỹ đạo có điểm số tốt nhất và từ đó đưa ra quyết định điều khiển tương ứng.

Với các tham số như trên, ta có thể viết tắt là một cấu hình config(M,N,K). Nếu $K = M$, hay tại mỗi bước ta đều xem xét N quỹ đạo con, ta sẽ phải tính toán tổng cộng N^M quỹ

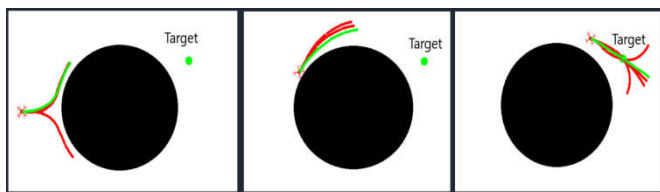
đạo. Trong thử nghiệm thực tế, K có thể nhỏ hơn nhiều so với M mà vẫn không làm giảm khả năng tránh vật cản của thuật toán. Ví dụ một cấu hình config(5, 3, 2). Nếu K = M, ta cần tính toán $3^5 = 243$ quỹ đạo, nhưng với K = 2, ta chỉ cần tính $3^2 = 9$ quỹ đạo. Với M, N lớn hơn, số quỹ đạo cần tính toán khi K = M sẽ rất lớn và tăng theo hàm mũ.



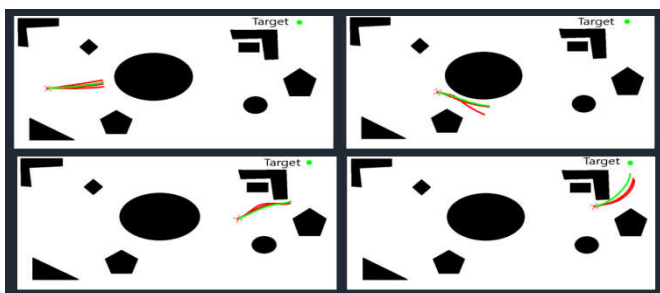
Hình 9. Ví dụ các quỹ đạo được tính toán với cấu hình config(4, 3, 1)

4. KẾT QUẢ MÔ PHỎNG

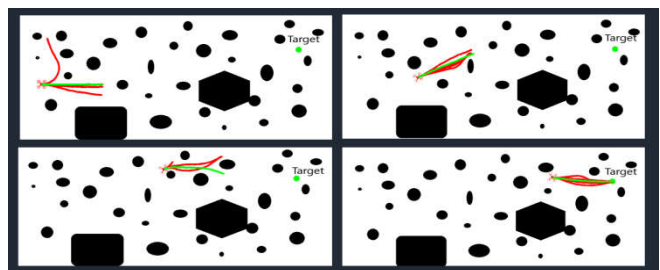
Thuật toán lập quỹ đạo di chuyển với bài toán điều khiển drone ở phần 3 được mô phỏng bằng ngôn ngữ Python kết hợp với sử dụng thư viện xử lý ảnh OpenCV. Cấu hình thuật toán config(4, 5, 1).



Hình 10. Mô phỏng thuật toán khi tránh vật cản kích thước lớn (điều kiện giống ngoài trời)



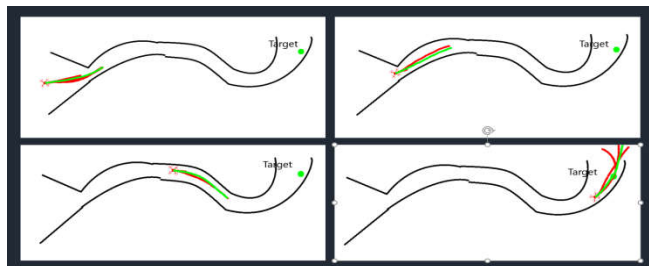
Hình 11. Mô phỏng thuật toán khi tránh vật cản kích thước nhỏ (điều kiện giống trong nhà)



Hình 12. Mô phỏng thuật toán khi tránh nhiều vật cản kích thước nhỏ (điều kiện giống trong rừng cây)

Ở tất cả các bài kiểm tra, thuật toán luôn tìm được đường đi an toàn đến mục tiêu. Có thể tăng các tham số M, N, K,

đồng thời giảm thời gian lấy mẫu để quỹ đạo di chuyển của drone là tối ưu hơn. Với cấu hình thuật toán config(4, 5, 1) thử nghiệm trên cấu hình phần cứng chip Intel core i5 8400, đạt fps 180; trên raspberry 4 đạt fps 58. Như vậy, thuật toán hoàn toàn đáp ứng thời gian thực với cấu hình phần cứng thấp. Đặc biệt, ta có thể cải thiện thuật toán thêm bằng cách sử dụng các kỹ thuật xử lý đa luồng hoặc dùng GPU tính toán các công đoạn xử lý ảnh.



Hình 13. Mô phỏng thuật toán khi đi trong không gian hẹp (đường hầm)

5. KẾT LUẬN

Bài báo đã đề xuất một thuật toán dùng để lập quỹ đạo di chuyển nhanh và đơn giản, đảm bảo an toàn trong điều kiện môi trường thay đổi cho các thiết bị tự hành. Thuật toán có độ phức tạp thấp, lập quỹ đạo thời gian thực ngay cả trên các máy tính nhúng giá rẻ như Raspberry Pi, Jetson Nano. Kết quả mô phỏng đối với bài toán điều khiển drone cho kết quả rất tốt trong nhiều dạng địa hình phức tạp. Quỹ đạo mà thuật toán tìm ra có thể trực tiếp thực hiện điều khiển đến drone mà không phải trải qua các quá trình mềm hóa quỹ đạo, giúp cho việc thiết lập điều khiển đơn giản hơn.

TÀI LIỆU THAM KHẢO

- [1]. Gregor Klancar, 2017. *Wheeled Mobile Robotics*. Elsevier Inc, pp. 161-206.
- [2]. Spyros G. Tzafestas, 2014. *Introduction to Mobile Robot Control*. Elsevier Inc, pp. 429-478.
- [3]. V.I. Zhulev, V.S. Leushkin, T.N. Nguyen, 2013. *Real-time trajectory planning for unmanned ground vehicle*. *Вестник ПФТУ*, pp. 18-22.
- [4]. The Open Motion Planning Library, Rice University [Online]. Available: <https://ompl.kavrakilab.org>
- [5]. Ioan A. Şucan, Mark Moll, Lydia E. Kavraki, 2012. *The Open Motion Planning Library*. *IEEE Robotics & Automation Magazine*, pp. 72-82.
- [6]. Zachary Kingston, Mark Moll, Lydia E, 2019. *Exploring Implicit Spaces for Constrained Sampling-Based Planning*. *International Journal of Robotics Research*, Houston, pp. 1151-1178.

AUTHORS INFORMATION

Nguyen Ngoc Tuan¹, Tang Thanh Lam¹, Lai Tien De¹, Tran Xuan Tinh²

¹Le Quy Don University (Military Technical Academy)

²Air Defence - Air Force Academy