

# Thử nghiệm thuật toán tìm đường đi tối ưu cho hệ thống robot di động vận chuyển hàng trong kho tự động

Nguyễn Chính Kiên<sup>1\*</sup>, Nguyễn Thị Hồng Hạnh<sup>1</sup>, Trương Xuân Hùng<sup>2</sup>

<sup>1</sup>Viện Cơ học, Viện Hàn lâm Khoa học và Công nghệ Việt Nam, 18 Hoàng Quốc Việt, phường Nghĩa Đô, Hà Nội, Việt Nam

<sup>2</sup>Trung tâm Vũ trụ Việt Nam, Viện Hàn lâm Khoa học và Công nghệ Việt Nam, 18 Hoàng Quốc Việt, phường Nghĩa Đô, Hà Nội, Việt Nam

Ngày nhận bài 9/12/2024; ngày chuyển phản biện 10/12/2024; ngày nhận phản biện 30/12/2024; ngày chấp nhận đăng 5/1/2025

## **Tóm tắt:**

Robot di động ngày càng được sử dụng phổ biến để vận chuyển hàng hóa trong kho hàng tự động. Nghiên cứu này trình bày việc áp dụng thử nghiệm thuật toán tìm đường đi tối ưu cho hệ thống robot di động vận chuyển hàng. Thuật toán được xây dựng và kiểm tra bằng mô hình giả lập, sau đó được áp dụng thử nghiệm để điều khiển hệ thống robot trong ba sơ đồ kho hàng thực tế theo thứ tự tăng dần về mức độ chi tiết, phức tạp do nhóm tác giả thiết lập. Một số cải tiến thuật toán đã được nghiên cứu khi gặp các vấn đề trong thực tế: quy tắc chiều đường đi, quy tắc ưu tiên thứ tự tại giao điểm, quy tắc chờ tại các điểm thắt nút (cửa kho). Trong quá trình thực nghiệm, robot vật lý chạy thực tế và cập nhật lộ trình di chuyển theo thời gian thực trên giao diện chương trình máy tính cho kết quả xử lý tốt với thời gian xử lý nhanh và quỹ đạo đã được kiểm chứng là đạt mức độ tối ưu vận hành, hoạt động ổn định với nhiều kịch bản nhiệm vụ. Kết quả này giúp cho những nhà nghiên cứu, nhà quản lý kho hàng có thể định hướng xây dựng mô hình phù hợp, tùy biến theo các điều kiện thực tiễn.

**Từ khóa:** robot di động, thuật toán A\*, thuật toán Dijkstra, tối ưu quãng đường, xe tự hành.

**Chỉ số phân loại:** 1.8, 2.2

## Experimenting with algorithms to find optimal paths for a mobile robot system transporting goods in an automated warehouse

Chinh Kien Nguyen<sup>1\*</sup>, Thi Hong Hanh Nguyen<sup>1</sup>, Xuan Hung Truong<sup>2</sup>

<sup>1</sup>Institute of Mechanics, Vietnam Academy of Science and Technology, 18 Hoang Quoc Viet Street, Nghia Do Ward, Hanoi, Vietnam

<sup>2</sup>Vietnam National Space Center, Vietnam Academy of Science and Technology, 18 Hoang Quoc Viet Street, Nghia Do Ward, Hanoi, Vietnam

Received 9 December 2024; revised 30 December 2024; accepted 5 January 2025

## **Abstract:**

Mobile robots are increasingly being used for goods transportation in automated warehouses. The study presents the experimental application of a routing algorithm to find the optimal paths for a mobile robot system transporting goods. The algorithms were first developed and tested in a simulation model, then experimentally applied to control the robot system in three warehouse layouts with increasing levels of detail and complexity, as designed by the research team. Several algorithmic improvements were studied to address practical issues, including path direction rules, priority at intersections, waiting rules at choke points (warehouse doors). During the experiments, the physical robot operated under real-world conditions and updated its movement trajectory in real-time on the computer program interface, which shows the efficient processing results with fast computation times, while the verified trajectory achieved an optimal level of operational performance and stable operation across multiple task scenarios. These results provide researchers and warehouse managers with insights to design and customise models according to practical conditions.

**Keywords:** A\* algorithm, automated guided vehicle, Dijkstra algorithm, mobile robot, optimal path.

**Classification numbers:** 1.8, 2.2

\*Tác giả liên hệ: Email: nckien@imech.vast.vn

## 1. Tổng quan

Ngày nay, trong kho hàng tự động, các robot tự hành phải thực hiện một lượng lớn hành trình vận chuyển hàng hóa trong một thời gian ngắn. Việc đưa ra lịch trình di chuyển của robot hợp lý để giảm chi phí vận hành thiết bị và thời gian chờ đợi là rất quan trọng. Vì vậy, việc nghiên cứu cách thức để đưa ra kế hoạch, phương pháp định tuyến tối ưu bằng chương trình máy tính là cần thiết.

Bài toán định tuyến này hướng tới việc thiết lập hành trình cho các robot trong kho mang các giá kệ hàng ra cửa kho và đưa chúng quay trở lại vị trí ban đầu để mà không vượt quá năng lực của mỗi robot với chi phí là nhỏ nhất. Bài toán định tuyến được đánh giá là một trong những thách thức lớn nhất trong lĩnh vực Vận trù học suốt 50 năm qua. Đây là một bài toán thuộc nhóm NP - khó, đã có nhiều nghiên cứu được tiến hành nhằm đề xuất các phương pháp giải quyết khác nhau như: thuật toán di truyền, thuật toán sinh cột hoặc kỹ thuật nhánh cận để tìm lời giải tối ưu.

E. Eade và cs (2007) [1] đã nghiên cứu một hệ thống định vị trong môi trường có phạm vi hạn chế, theo phương pháp tiếp cận dựa trên một camera đơn lẻ, quy mô thử nghiệm được xây dựng thành các nút riêng biệt, kết nối với nhau thông qua một đồ thị lớn bằng cách áp dụng các kỹ thuật tối ưu hóa phi tuyến. F. Dellaert và cs (2006) [2], H. Strasdat và cs (2010) [3] đã sử dụng các kỹ thuật như Smoothing, Bundle Adjustment (BA) và Mapping (SAM) để theo dõi các đặc trưng qua từng khung hình trong chuỗi ảnh liên tiếp.

Nghiên cứu của B. Williams (2009) [4] đã thực hiện nghiên cứu đánh giá tầm quan trọng của thủ tục vòng lặp đóng, giúp kiểm tra xem các vị trí trong hệ thống SLAM sử dụng một camera đơn lẻ (Monocular SLAM) đã được nhận diện hay chưa. Phương pháp này được sử dụng cho các bài toán định vị sử dụng hình ảnh và sử dụng ba phương pháp đối sánh ảnh là: Map-to-Map [5], Image-to-Map [6] và Image-to-Image [7, 8]. Thử nghiệm được tiến hành trong môi trường trong nhà, ngoài trời, với quỹ đạo di chuyển là một vòng đơn và nhiều vòng. Kết quả cho thấy, đối sánh Image-to-Image đạt hiệu suất cao nhất, trong khi Map-to-Map và Image-to-Map có hiệu quả thấp hơn.

S. Perera và cs (2011) [9] đã phát triển một thuật toán giám sát nhằm theo dõi các đối tượng chuyển động trong môi trường. Thuật toán này giải quyết vấn đề nhận diện các điểm chuyển động từ một camera đang di chuyển, dựa trên các ràng buộc epipolar và thông tin được thu thập từ thuật toán monoSLAM. Tuy nhiên, để thiết lập ngưỡng phân loại chính xác cho các điểm chuyển động, cần thực hiện nhiều thử nghiệm, phụ thuộc vào số lượng và phân bố của các điểm chuyển động được phát hiện trong môi trường nghiên cứu.

R. Ozawa và cs (2005) [10] đã phát triển một hệ thống trực tuyến nhằm tái tạo bản đồ 3D thông qua kỹ thuật đo hành trình dựa trên dữ liệu thị giác (visual odometry). Hệ thống này bao gồm 2 thành phần: thành phần đầu tiên xử lý việc tạo dựng

bản đồ 3D từ chuỗi hình ảnh thu được từ camera-stereo, giúp mô tả chi tiết môi trường thực tế và hỗ trợ việc lập kế hoạch di chuyển cho robot trên bản đồ đã được tái tạo; thành phần thứ hai đo hành trình thị giác nhằm kết nối các khung hình 3D liên tiếp để xác định mô hình chuyển động của camera theo phương pháp sáu bậc tự do (6DOF) cùng với thông tin về môi trường 3D. Các kỹ thuật sử dụng bao gồm: tính toán độ sâu của hình ảnh, ước lượng chuyển động camera theo mô hình 6DOF bằng kỹ thuật RANSAC [11]. Ngoài ra, dữ liệu kết quả từ bản đồ 3D được sử dụng để lập kế hoạch di chuyển cho robot bằng cách kết hợp các điểm dự đoán cho những bước di chuyển tiếp theo. P. Michel và cs (2007) [12] đã đề xuất một phương pháp thời gian thực trên GPU để hỗ trợ robot di chuyển và leo cầu thang trong không gian 3D, dựa trên chuỗi hình ảnh thu được từ camera-stereo. Tuy nhiên, phương pháp này có giới hạn nhất định là sự lệ thuộc với đối tượng 3D cần theo dõi và mô hình 3D có kích thước nhỏ, nhưng nó đặc biệt hiệu quả trong các tình huống cụ thể như leo cầu thang.

Trong nghiên cứu của H.T. Tran và cs (2022) [13], phương pháp Gradient cùng bộ điều khiển mờ Takagi-Sugeno đã được sử dụng để tối ưu hóa các tham số, nhằm điều chỉnh quá trình điều hướng và tránh chướng ngại vật cho robot di động. Kiến trúc điều khiển mờ này mô phỏng hành vi của robot trong môi trường chưa biết, ở đây bao gồm các hành vi như tìm kiếm mục tiêu và tránh vật cản, theo dõi robot. Robot di động bánh xe có trang bị bộ điều khiển logic mờ được thực hiện trên vi điều khiển Atmega, với dữ liệu đầu vào từ các cảm biến kiểm soát động cơ bên trái và bên phải. Ngoài ra, việc sử dụng bộ điều khiển mờ kết hợp với thuật toán di truyền đã được triển khai cho một nhóm robot tự hành trong nhà. Trong đó, bộ điều khiển mờ hỗ trợ một robot ban đầu, sau đó thuật toán di truyền tối ưu hóa chức năng điều khiển khi di chuyển trong các môi trường phức tạp.

Q. Liu và cs (2006) [14] trong nghiên cứu của mình đã sử dụng thuật toán di truyền để điều khiển robot tự hành (AMR), kết hợp dữ liệu từ cảm biến siêu âm nhằm xác định các thông số môi trường và lập kế hoạch đường đi. So sánh giữa hai loại mô hình mờ Takagi-Sugeno và Mamdani cho thấy, mô hình Mamdani mang lại sự vận hành mượt mà hơn, trong khi Takagi-Sugeno lại tiêu thụ ít tài nguyên bộ nhớ hơn, phù hợp khi sử dụng với các vi điều khiển trong thời gian thực.

Nhiều nghiên cứu kết hợp thuật toán logic mờ với các phương pháp tính toán từ phần mềm, bao gồm thuật toán di truyền, mạng nơ-ron nhân tạo và tối ưu hóa bầy đàn hạt, nhằm tối ưu hóa các tham số trong bộ điều khiển mờ và cải thiện được hiệu suất điều hướng cho robot di động [15]. Các hành vi logic mờ cơ bản được thiết kế trong các nghiên cứu này bao gồm: di chuyển theo mục tiêu và tránh các chướng ngại vật. Kết hợp các thuật toán này, robot có thể lập kế hoạch đường đi hiệu quả trong môi trường có các chướng ngại vật động. Thuật toán di truyền và mạng nơ-ron cũng đã được áp dụng để cải thiện khả năng điều hướng của robot. Các thuật toán tiên tiến được thiết kế nhằm tối ưu hóa tham số đầu vào và hiệu quả hoạt

động của bộ điều khiển mờ, nâng cao hiệu suất trong các môi trường phức tạp. Hành vi tránh vật cản được điều chỉnh nhờ dữ liệu từ cảm biến phạm vi để xác định khoảng cách tới chướng ngại vật gần nhất, trong khi hành vi tìm kiếm mục tiêu dựa vào cảm biến la bàn để xác định hướng di chuyển chính xác. Trong nghiên cứu của M. Faisal và cs (2013) [16], thuật toán di truyền mờ (GA-FL) được áp dụng để giải quyết bài toán lập kế hoạch chuyển động trong môi trường động. Nghiên cứu sử dụng các cảm biến không dây để hỗ trợ robot di chuyển trong các môi trường chứa cả chướng ngại vật tĩnh và động. Hai bộ điều khiển mờ được thiết kế, bao gồm điều khiển theo dõi logic mờ (TFLC) và điều khiển tránh vật cản logic mờ (AFLC) để đảm bảo robot có thể tìm kiếm đường đi tối ưu từ điểm xuất phát tới mục tiêu trong nhiều tình huống vận hành khác nhau. Ngoài ra, nghiên cứu cũng đề xuất phương pháp lập kế hoạch đường đi trực tuyến theo nguồn dữ liệu thu được từ cảm biến trong môi trường động. Các bộ điều khiển mờ được thiết kế để hoạt động trong thời gian thực sử dụng vi mạch (FPGA) có thể lập trình lại. Các luật mờ thực hiện trong nghiên cứu này được xây dựng nhằm mô phỏng hành vi lái xe của con người và đã được kiểm chứng thông qua các hệ thống tác nhân. Trong đó, robot có khả năng tự học hỏi và điều hướng hiệu quả ngay cả trong các môi trường phức tạp và không chắc chắn.

H. Xiao và cs (2007) [17] đã kết hợp các lớp mạng nơ-ron nhân tạo để tạo kho dữ liệu đầu vào nhân tạo. Đồng thời, phương pháp học tăng cường Q được sử dụng để phát triển thuật toán lập kế hoạch đường đi cho robot di động. Hệ thống này điều khiển tốc độ động cơ DC của robot qua vi điều khiển Arduino Uno, với sự hỗ trợ của nhiều lớp nơ-ron. Với bộ điều khiển mạng nơ-ron và PID được xây dựng và thử nghiệm trên các robot có mang hộp số tự động. Một số tác giả khác đã sử dụng mạng nơ-ron để tạo ra các đường đi dẫn tránh va chạm trong môi trường không xác định, đặc biệt là trong các tình huống yêu cầu khả năng lập kế hoạch trực tuyến khi robot hoạt động trong môi trường có chướng ngại vật không được biết trước.

Một số nghiên cứu khác đã phân tích các hành vi tránh chướng ngại vật sử dụng kết hợp mạng nơ-ron và học tăng cường. Trong nghiên cứu, C.T. Vo và cs (2021) [18] đã phát triển bộ điều khiển mạng nơ-ron chuyên tiếp sử dụng nguồn cấp dữ liệu đa lớp, để tự động điều khiển hướng lái của robot trong các môi trường tĩnh và động. Mạng nơ-ron này có bốn lớp đầu vào dựa trên khoảng cách đi tới chướng ngại vật và đầu ra là góc lái của các robot. Tuy vậy, ở môi trường năng động và không có cấu trúc, xác định đường dẫn phù hợp để giảm thiểu va chạm trở nên phức tạp hơn đáng kể.

Nghiên cứu của S.X. Yang và cs (2003) [19] đã giới thiệu giải pháp nhằm lập kế hoạch đường đi cho robot di động và bản địa hóa cho robot di động bằng cách sử dụng mạng thần kinh nhân tạo hồi quy (RNN). Nghiên cứu áp dụng hai bộ điều khiển mạng thần kinh cho mục tiêu lập kế hoạch và không chế đường

đẫn: một bộ điều khiển hỗ trợ robot phát hiện các khoảng trống trong không gian. Bộ điều khiển thứ hai giúp huấn luyện robot nhằm tránh chướng ngại vật. Hệ thống dùng cảm biến tầm nhìn loại Kohonen, được thiết kế chuyên dụng cho việc điều hướng robot thông minh.

Trong nghiên cứu của A. Zhu và cs (2007) [20], tác giả đã đề xuất một phương pháp nhằm điều hướng dựa trên lý thuyết của điều khiển mờ, tận dụng phản ứng của các robot ở trong môi trường không xác định. Hệ thống này sử dụng 48 luật mờ để điều khiển hai hành vi chính: tìm kiếm mục tiêu và tránh chướng ngại vật. Các thông số chức năng của mạng nơ-ron được điều chỉnh để cải thiện khả năng học tập của robot, giúp rút ngắn đường đi từ điểm xuất phát đến đích. Cảm biến siêu âm cung cấp dữ liệu về các chướng ngại vật xung quanh, làm đầu vào cho bộ điều khiển mờ, đầu ra của bộ điều khiển sẽ kiểm soát tốc độ của hai bánh xe trái, phải. Trong mô hình này, bộ điều khiển mờ và mạng nơ-ron nhân tạo sử dụng hàm cơ sở xuyên tâm để kiểm soát vận tốc và điều chỉnh sự biến đổi môi trường.

S.U. Deshpande và cs (2013) [21] đã xây dựng một mạng nơ-ron mờ dựa trên kinh nghiệm, sử dụng dữ liệu từ cảm biến siêu âm để xác định tốc độ của robot. Nghiên cứu áp dụng 16 quy tắc mờ để kiểm soát hướng di chuyển của robot, với hai hệ thống chính là hệ thống thần kinh mờ thích ứng (ANFIS) và hệ thống thần kinh mờ lai. Bộ điều khiển mờ này dựa vào dữ liệu từ cảm biến giúp robot di chuyển an toàn trong môi trường trong nhà. Nghiên cứu cũng chỉ ra rằng, sự kết hợp giữa mạng nơ-ron và bộ điều khiển mờ có thể nâng cao hiệu quả trong việc điều hướng, kiểm soát hướng đi của robot.

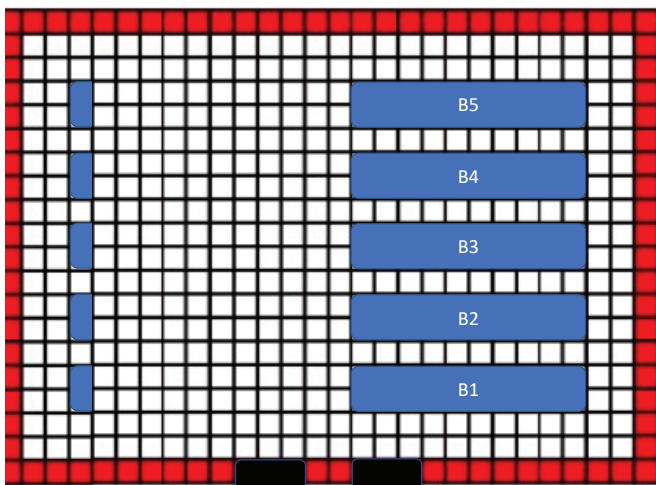
Nghiên cứu của M. Gao và cs (2007) [22] đã phát triển một hệ thống tối ưu cho bộ điều khiển logic mờ, nhằm hỗ trợ robot di động trong việc điều hướng và tránh vật cản. Phương pháp tối ưu này kết hợp thuật toán di truyền với thuật toán ủ mô phỏng (Simulated annealing algorithm - SAA). Thuật toán ủ mô phỏng và bộ điều khiển thông minh được thiết kế để robot xác định lộ trình tối ưu trong môi trường phức tạp. Ngoài ra, việc tích hợp thuật toán ủ mô phỏng với tối ưu hóa đàn kiến (ACO) đã cải thiện đáng kể tốc độ điều hướng của robot, trong khi việc kết hợp thuật toán ủ mô phỏng và mạng nơ-ron lại giúp cải thiện về tốc độ hội tụ.

Bên cạnh đó, nhiều nghiên cứu khác cũng đề xuất giải pháp xây dựng kế hoạch điều hướng dành cho robot di động thông qua việc ứng dụng thuật toán ACO và thuật toán Dijkstra [23]. Thuật toán ACO được sử dụng để tìm kiếm các giá trị tối ưu được lấy từ bảng luật mờ, nhằm giảm khoảng cách di chuyển từ điểm xuất phát đến điểm đích của robot, đồng thời hỗ trợ robot hiệu quả trong việc tránh chướng ngại vật. Các cải tiến từ ACO không chỉ tối ưu hóa tốc độ tìm đường của robot di động mà đáp ứng linh hoạt các hành vi như tìm kiếm mục tiêu và tránh vật cản.

Ở Việt Nam, bài toán này cũng được nghiên cứu sâu rộng từ lý thuyết đến ứng dụng vào thực tế bởi các tổ chức nghiên cứu, trường đại học cũng như các công ty logistics. Nhóm các nhà nghiên cứu của Học viện Kỹ thuật Quân sự đã chế tạo thành công hệ thống robot y tế vận chuyển Vibot vào năm 2023 [24], tuy nhiên, vẫn tồn tại một số thách thức trong việc phát triển các hệ thống này, bao gồm: Độ chính xác của các hệ thống định vị và dẫn đường, độ tin cậy và độ bền của các cảm biến, phải đảm bảo hệ thống có thể hoạt động ổn định trong môi trường thay đổi và không đồng nhất. Bên cạnh đó, việc tích hợp các cảm biến và xây dựng hệ thống truyền thông cho phép quan sát mục tiêu từ xa thông qua hệ thống camera được trang bị trên robot ở chế độ thời gian thực cũng là một thách thức trong lĩnh vực nghiên cứu.

N.K. Nguyen và cs (2023) [25] đã xây dựng thuật toán định tuyến mới dựa trên cơ sở giải thuật Dijkstra áp dụng cho các robot tự hành trong hệ thống chia chọn bưu phẩm, bưu kiện. Đặc biệt, thuật toán chú trọng tới các yếu tố thay đổi trong quá trình hoạt động của hệ thống, từ đó điều chỉnh một cách đồng bộ để mang lại quỹ đạo tối ưu cho từng robot, cũng như tối thiểu chi phí của toàn hệ thống và đã được áp dụng trong hệ thống kho của Viettel Post. Nghiên cứu này đã thiết lập quy tắc: Sau khi robot thứ nhất xác định đường đi, robot thứ hai coi các nút trên đường đi của robot 1 là vật cản và thay đổi trọng số để thuật toán xác định lại đường đi của robot 2 (tránh các nút đó), tiếp tục cho các robot sau nữa. Tuy nhiên, khi kho hàng nhỏ, có số lượng đường đi ít (tối ưu hóa không gian chứa hàng), đặc biệt là các quỹ đạo đường đi đều hướng ra cửa kho nên phương án xác định này là không phù hợp thực tế.

Trong nghiên cứu này, các tác giả thử nghiệm một số thuật toán cho bài toán hoạch định tuyến tối ưu của các robot tải hàng trong nhà kho. Các thuật toán sẽ được thử nghiệm với bài toán giả lập để đánh giá hiệu quả, sau đó sẽ được thử nghiệm trên mô hình nhà kho thật với một số điều kiện: chiều đi giữa các nút, trạng thái tải/không tải của robot, thứ tự xếp hàng chờ tại cửa kho... (hình 1).



Hình 1. Sơ đồ kho hàng.

## 2. Đối tượng và phương pháp nghiên cứu

Sơ đồ mặt bằng của kho hàng được tổ chức thành một bản đồ, hình 1 minh họa một sơ đồ mặt bằng cụ thể. Theo đó, bản đồ có các thông tin về vị trí của giá để hàng hoá, được tập hợp thành dãy. Các dãy giá hàng được bố trí cách đều nhau, khoảng không gian giữa các dãy tạo thành con đường có thể di chuyển được. Thông qua tọa độ hóa vị trí trên bản đồ bằng cách gán giá trị định vị cho từng nút gắn trên sơ đồ mặt bằng. Khi đó, các con đường (được gọi là cạnh) kết nối các nút; mỗi nút có thể kết nối tới một hoặc nhiều cạnh. Giá thiết mỗi cạnh có tương ứng một giá trị chi phí, ví dụ độ dài hoặc thời gian cần thiết để di chuyển cạnh đó. Đối với những bản đồ có kích thước nhỏ và phương án bố trí giá hàng đơn giản, việc tính toán tất cả các tuyến đường có thể đến đích và lựa chọn tuyến đường ngắn nhất là khả thi. Tuy nhiên, khi kích thước của bản đồ tăng lên cùng với sự phức tạp trong phương án bố trí giá hàng làm cho quá trình tính toán cho bản đồ có số lượng nút lớn tăng theo cấp số nhân. Do đó, thuật toán tìm đường đi tối ưu được nghiên cứu để áp dụng trong thực tế cho những mặt bằng kho hàng loại này.

### 2.1. Thuật toán Dijkstra

Năm 1959, E.W. Dijkstra đã giới thiệu thuật toán Dijkstra, đây là một phương pháp để giải quyết bài toán tìm đường đi ngắn nhất từ nguồn đơn trong một đồ thị có hướng với điều kiện không có cạnh mang trọng số âm. Thuật toán hoạt động giá trị gần đúng bằng cách liên tục cập nhật khoảng cách chính xác hơn. Ban đầu, khoảng cách đến mỗi đỉnh được ước tính là giá trị lớn hơn thực tế. Qua mỗi bước, giá trị này được thay thế bằng giá trị nhỏ hơn nếu tìm được đường đi ngắn hơn.

Quy trình hoạt động của thuật toán Dijkstra bao gồm các bước duyệt qua các đỉnh của đồ thị lần lượt theo khoảng cách ngắn nhất từ đỉnh xuất phát, mở rộng từ đỉnh này đến các đỉnh kế tiếp theo có chi phí thấp nhất, đồng thời duy trì một bộ đường đi ngắn nhất tới các đỉnh đã xem xét.

Thuật toán sử dụng hàng đợi ưu tiên kết hợp với phương pháp tham lam để chọn đỉnh gần nhất chưa được xử lý, sau đó thực hiện quá trình giảm bớt trên tất cả các cạnh liên quan. Nguyên lý hoạt động của thuật toán bao gồm các bước sau [23]:

*Khởi tạo:* Đặt giá trị khoảng cách từ đỉnh xuất phát là 0; gán giá trị khoảng cách là  $+\infty$  cho tất cả các đỉnh còn lại; tạo một tập hợp trống để lưu trữ các đỉnh đã được xử lý.

*Lặp lại:* Tìm đỉnh chưa được xử lý có khoảng cách là nhỏ nhất (đỉnh xuất phát ở bước đầu); cập nhật khoảng cách cho các đỉnh kề: nếu khoảng cách từ đỉnh hiện tại qua một cạnh đến đỉnh kề bé hơn khoảng cách đã biết, cập nhật lại giá trị khoảng cách này; đánh dấu ở đỉnh hiện tại là đã được xử lý.

*Kết thúc:* Quá trình tiếp tục được lặp lại cho đến khi tất cả các đỉnh đã được xử lý hoặc khi tìm được khoảng cách ngắn nhất đến đích; dựa vào nguyên lý hoạt động của thuật toán Dijkstra để tìm đường đi ngắn nhất từ đỉnh A đến các đỉnh

khác, các tác giả đã lập trình theo quy trình từng bước dựa trên nguyên lý hoạt động trên của thuật toán: từ nút bắt đầu, thêm tất cả các nút được kết nối vào hàng ưu tiên; sắp xếp hàng đợi ưu tiên theo chi phí thấp nhất và đặt nút đầu tiên làm nút hiện tại; đối với mỗi nút con, chọn nút tốt nhất dẫn đến con đường ngắn nhất để bắt đầu; khi tất cả các cạnh đã được tính từ một nút, nút đó gán đã duyệt và không cần phải đến đó nữa; thêm từng nút con được kết nối với nút hiện tại vào hàng ưu tiên; chuyển sang bước 2 cho đến khi hàng đợi trống; đệ quy tạo danh sách từng nút nút dẫn đường đi ngắn nhất từ đầu đến cuối; đảo ngược danh sách và bạn đã tìm thấy con đường ngắn nhất.

Nói cách khác, theo phương pháp đệ quy, đối với mỗi nút kế tiếp của một nút, thực hiện đo khoảng cách giữa chúng đến điểm bắt đầu, so sánh và lưu trữ khoảng cách cùng với nút tương ứng với con đường ngắn nhất. Khi đến nút cuối, đệ quy quay lại điểm xuất phát theo con đường ngắn nhất, đảo ngược danh sách đó và có được đường đi ngắn nhất.

## 2.2. Thuật toán A\*

Kể từ khi thuật toán Dijkstra được công bố, nhiều giải pháp cải tiến thuật toán này đã được đề xuất và A\* là một thuật toán phổ biến nhất. A\* là một thuật toán tìm kiếm theo cơ chế heuristic, tức là nó sử dụng thông tin dự đoán để hướng dẫn quá trình tìm kiếm. Thuật toán này kết hợp giữa tìm kiếm theo chi phí thấp nhất (thuật toán Dijkstra) và tìm theo chiều sâu, giúp tìm ra đường đi ngắn nhất một cách nhanh chóng.

A\* đã có một sửa đổi đơn giản, cụ thể là các cạnh cũng được ưu tiên dựa trên mức độ gần hơn của cạnh đó dẫn đến khoảng cách theo đường thẳng tới mục tiêu. Vì vậy, trước khi chạy tìm kiếm A\*, khoảng cách đường thẳng đến đích cuối cùng phải được đo cho mọi nút, điều này thật dễ dàng nếu biết tọa độ của từng nút. Đây là dạng đơn giản nhất của A\* và định nghĩa của nó cũng cho phép cải tiến chức năng phỏng đoán. Thuật toán này có lợi thế lớn về hiệu suất vì nó không cần phải truy cập nhiều nút khi biết hướng cuối cùng của đường dẫn.

A\* hoạt động bằng cách mở rộng các nút (điểm) từ điểm bắt đầu đến điểm kết thúc dựa trên hàm đánh giá  $f(n)$ . Hàm này kết hợp giữa chi phí đi từ điểm bắt đầu đến điểm hiện tại ( $g(n)$ ) và có chi phí ước lượng từ điểm hiện tại đến điểm đích ( $h(n)$ ) như sau:

$$f(n) = g(n) + h(n)$$

trong đó:  $g(n)$  là chi phí từ điểm bắt đầu đến điểm hiện tại;  $h(n)$  là hàm heuristic cho ước lượng chi phí từ điểm hiện tại đến điểm đích.

Nguyên lý của thuật toán A\* được mô tả như sau:

**Khởi tạo:** Đặt điểm bắt đầu vào danh sách mở (open list) và gán giá trị  $f$  ban đầu cho nó; đặt danh sách đóng (closed list) rỗng.

**Lập lại:** Chọn nút trong danh sách mở có giá trị  $f$  nhỏ nhất và di chuyển nó sang danh sách đóng; nếu nút này là điểm đích, thì đường đi ngắn nhất đã được tìm thấy.

Nếu không, đối với mỗi nút kề cận: Tính giá trị  $g$  và  $f$  cho nút kề; nếu nút này chưa có trong danh sách mở hoặc có giá trị  $f$  nhỏ hơn giá trị hiện tại trong danh sách mở, thì cập nhật giá trị và đặt nút này vào danh sách mở.

**Kết thúc:** Lập lại quá trình cho đến khi tìm thấy điểm đích hoặc danh sách mở rỗng (đường đi không tồn tại).

Các tác giả đã lập trình theo quy trình dựa trên nguyên lý hoạt động trên của thuật toán A\* [23] như sau: Khởi tạo đồ thị và tạo một hàng đợi ưu tiên để lưu trữ các đỉnh chưa được xét đến; đặt các giá trị ban đầu cho toàn bộ đồ thị là vô cùng và chọn đỉnh bắt đầu; tính toán giá trị  $f(g+h)$  cho mỗi đỉnh, trong đó:  $g$  là khoảng cách từ đỉnh bắt đầu đến đỉnh đó và  $h$  là giá trị heuristic từ đỉnh đó đến đỉnh kết thúc; đặt đỉnh bắt đầu vào hàng đợi ưu tiên; trong khi hàng đợi không rỗng, lặp lại các bước sau: lấy đỉnh có  $f$  nhỏ nhất ra khỏi hàng đợi ưu tiên và đặt là đỉnh hiện tại; nếu đỉnh hiện tại là đỉnh kết thúc, dừng thuật toán; đánh dấu đỉnh hiện tại là đã xét; tìm các đỉnh kề của đỉnh hiện tại và tính toán giá trị  $f$  cho chúng; nếu đỉnh kề chưa được xét, đặt giá trị  $f$  và đẩy nó vào hàng đợi ưu tiên; trả về đường đi tới đỉnh kết thúc.

## 2.3. Xây dựng hệ thống điều khiển cho hệ nhiều robot

Đối với việc xác định đường đi của một robot đơn lẻ trong một không gian xác định, các thuật toán Dijkstra và A\* đã xác định tốt được quỹ đạo di chuyển tối ưu theo điều kiện “chi phí”. Tuy nhiên, khi có nhiều robot cùng làm việc trong không gian, đường đi của từng robot thường xuyên xảy ra chông chéo, cản trở hoặc đôi khi đối đầu nhau. Điều này làm cho các robot phải có khả năng xử lý tránh nhau và tiếp tục đường đi ban đầu, dẫn đến việc phải tính toán, xử lý quỹ đạo trên từng robot đòi hỏi yêu cầu lớn hơn về tài nguyên và kết quả thu được đôi khi không đảm bảo tính tối ưu toàn cục cho toàn hệ robot. Để hạn chế những xung đột nêu trên, cần đặt ra các quy tắc vận chuyển của từng robot tại các giao điểm, quy tắc vận chuyển theo các chiều đi. Giải pháp cụ thể được nhóm tác giả đề xuất để kết hợp với thuật toán tìm đường tối ưu cho hệ nhiều robot trong bài toán thực như sau: Khi nhận được yêu cầu xử lý một đơn hàng, ví dụ lấy nhiều loại hàng từ nhiều giá hàng ở những vị trí khác nhau đưa ra cửa kho. Xác định nhiệm vụ cho từng robot dựa trên chi phí tổng thể toàn bộ công việc dựa vào trạng thái, vị trí tức thời của từng robot; bản đồ đường đi khi robot đang mang giá hàng và không mang giá hàng là khác nhau, theo đó, robot không mang giá hàng có thể di chuyển trong vùng không gian bên dưới giá hàng. Đường đi của robot thực thi tác vụ được phân đoạn với trọng số di chuyển khác nhau. Thực hiện tìm thuật toán tối ưu cho từng phân đoạn; các hành lang có tính định hướng chiều di chuyển, trọng số nhỏ theo đúng chiều và trọng số rất lớn nếu theo chiều ngược lại; thiết lập dãy chờ tại cửa kho cùng với quy tắc chờ lần lượt: vận tốc robot giảm và chờ di chuyển theo trạng thái robot xếp hàng phía trước; khi có từ hai robot trở lên tại điểm giao nút trong cùng một thời điểm, xác định thứ tự di chuyển theo phân mức ưu tiên của chúng.

Hai thuật toán Dijkstra và A\* được sử dụng nhằm tìm ra đường đi tối ưu cho các robot, sử dụng bộ quy tắc 5 điểm để làm điều kiện vận hành đối với hai thuật toán. Xét một cách tổng quát với vị trí xuất phát là  $(q_{10}, q_{20}, \dots, q_{N0})$ , vị trí đích là  $(q_{1G}, q_{2G}, \dots, q_{NG})$ . Các vị trí nêu trên đều biết trước, trong đó vị trí xuất phát được xác định dựa trên dữ liệu di chuyển của bản thân từng robot và vị trí đích được xác định từ tác vụ mà robot được phân công, vị trí giá hàng cần mang.

### 3. Mô hình kho hàng và tính toán

#### 3.1. Thiết lập kho hàng và sơ đồ khối của hệ thống điều khiển

Nhóm tác giả thiết lập 3 sơ đồ kho hàng theo thứ tự tăng dần về mức độ chi tiết, phức tạp. Sơ đồ mặt bằng thứ nhất bao gồm 8 ô có kích thước 1x1 m, chia thành 2 hàng, mỗi hàng có 4 ô; mỗi ô được gá mã định danh duy nhất. Số lượng robot sử dụng là 1. Hình 2 minh họa hình ảnh thực tế của sơ đồ mặt bằng thứ nhất.



Hình 2. Sơ đồ mặt bằng thứ nhất.

Sơ đồ mặt bằng thứ hai bao gồm 20 ô 1x1 m, chia thành 4 hàng, 1 hàng có 4 ô và 3 hàng có 5 ô/hàng. Số lượng robot sử dụng là 2 chiếc cùng với 4 giá hàng. Hình 3 minh họa hình ảnh thực tế của sơ đồ mặt bằng thứ hai.



Hình 3. Sơ đồ mặt bằng thứ hai.

Sơ đồ mặt bằng thứ ba có diện tích lớn với số lượng robot sử dụng là 10 chiếc. Mặt bằng hình chữ nhật, có kích thước 26x18 ô 1x1 m, tương ứng với diện tích tối thiểu là 468 m<sup>2</sup>. Mặt bằng cho phép bố trí 20 dãy giá hàng với tổng số 200 giá. Kho hàng này được thử nghiệm thực thi nhiều đơn hàng (nhiều nhiệm vụ) khác nhau ngẫu nhiên cho phù hợp với bài toán diễn ra trong thực tế.

Quy trình xử lý một đơn hàng, tương ứng với một tác vụ gán cho robot, được mô tả gồm 5 bước như sau:

*Bước 1:* Cần xác định số lượng giá hàng cần vận chuyển đến cửa kho, sử dụng giải thuật phân công tác vụ để xác định số lượng robot cần tham gia vào xử lý đơn hàng. Lập tổ hợp cặp robot - giá hàng để tìm ra tổng đường đi ngắn nhất sử dụng một trong hai thuật toán Dijkstra hoặc A\*. Xác lập cặp robot - giá hàng.

*Bước 2:* Gửi quyết định thực thi cho từng robot di chuyển đến vị trí giá hàng tương ứng. Đường đi của các robot có thể nằm một phần ở không gian bên dưới các giá hàng, điều kiện ràng buộc là khu vực có vật cản tĩnh và chiều di chuyển tại hành lang.

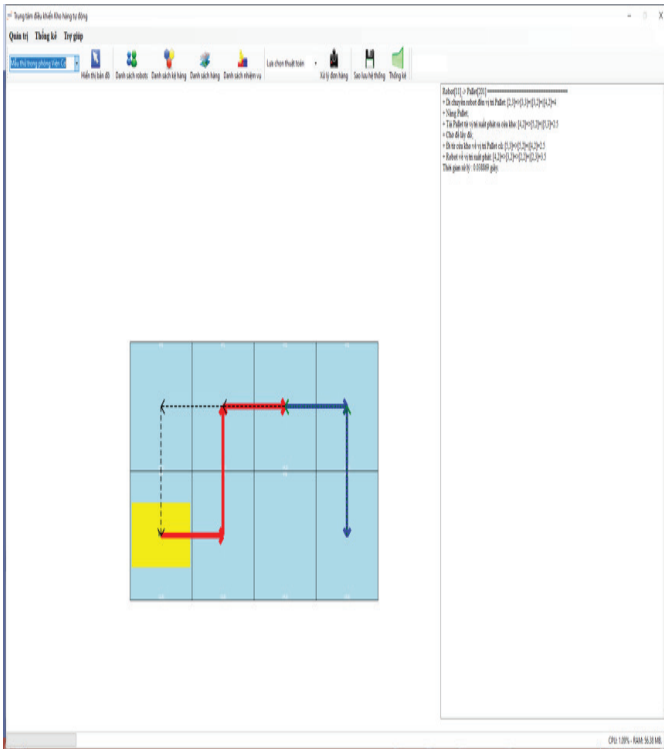
*Bước 3:* Robot mang giá hàng ra cửa kho. Khi đó, đường đi sẽ là các hành lang, chính là toàn bộ bản đồ loại trừ các khu vực có vật cản tĩnh và khu vực xếp giá hàng. Quá trình di chuyển, bản đồ được cập nhật thông tin về vị trí hiện tại của tất cả robot và thông tin khác có liên quan.

*Bước 4:* Robot mang giá hàng từ cửa kho về vị trí đặt giá hàng ban đầu. Khi đó, đường đi sẽ là các hành lang, chính là toàn bộ bản đồ loại trừ các khu vực có vật cản tĩnh và khu vực xếp giá hàng. Trong quá trình di chuyển, bản đồ có thể được cập nhật thông tin có liên quan nếu cần thiết.

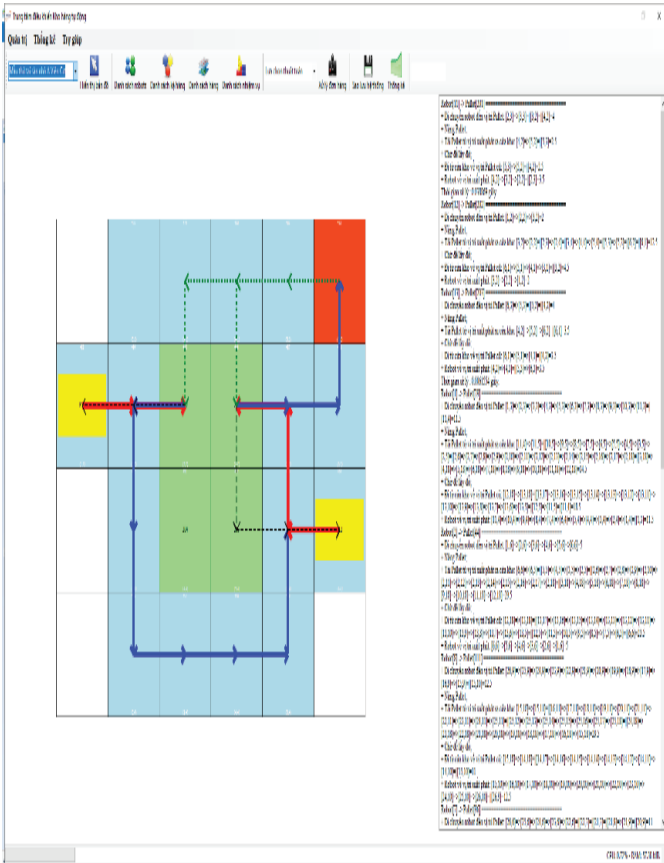
*Bước 5:* Robot quay trở về vị trí ban đầu. Đường đi của robot có thể nằm một phần ở không gian bên dưới các giá hàng, điều kiện ràng buộc là khu vực có vật cản tĩnh và chiều di chuyển tại hành lang. Bản đồ được cập nhật thông tin về vị trí hiện tại của tất cả robot và thông tin khác có liên quan.

#### 3.2. Kết quả thử nghiệm và bàn luận

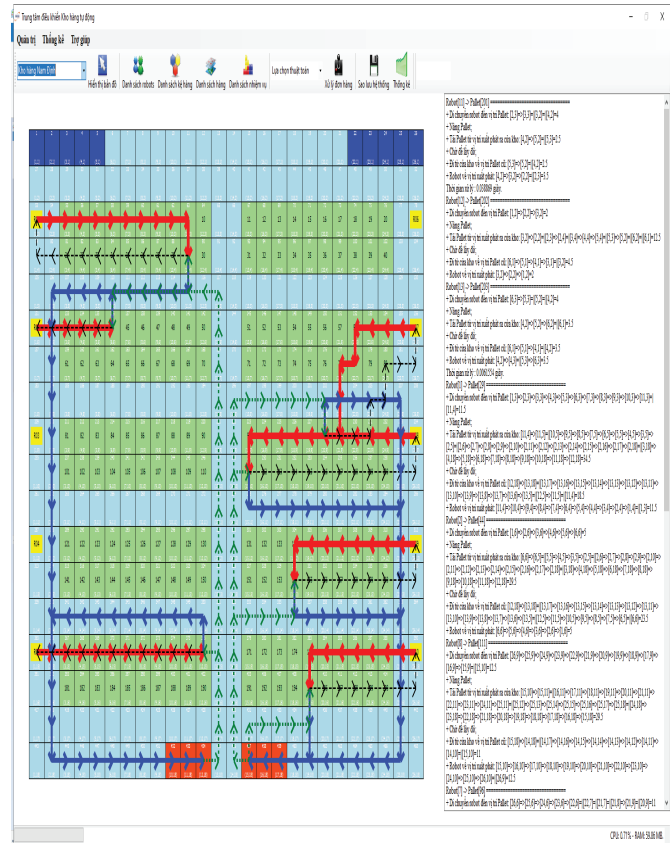
Thử nghiệm hai thuật toán với giải pháp điều khiển và quy trình xử lý đơn hàng cho cả ba sơ đồ mặt bằng. Hệ thống đã vận hành tốt, thuật toán đã đưa ra kết quả lộ trình hợp lý cho từng loại sơ đồ, minh họa trong các hình 4-6. Do đặc điểm bố trí giá hàng trong kho hàng đơn giản, kích thước mặt bằng không quá lớn nên các chỉ tiêu hoạt động của hai thuật toán gần như không có sai lệch, cụ thể được trình bày trong bảng 1.



Hình 4. Xác định đường đi tối ưu trong sơ đồ mặt bằng thứ nhất.



Hình 5. Xác định đường đi tối ưu trong sơ đồ mặt bằng thứ hai.



Hình 6. Xác định đường đi tối ưu trong sơ đồ mặt bằng thứ ba.

Bảng 1. Kết quả chỉ số đánh giá mô hình huấn luyện của hai thuật toán.

Bản đồ	Thuật toán Dijkstra		Thuật toán A*	
	Quãng đường (m)	Thời gian xử lý (giây)	Quãng đường (m)	Thời gian xử lý (giây)
Kho hàng 1	12,5	0,0307606	12	0,030702
Kho hàng 2	45,5	0,0480335	44,5	0,0356332
Kho hàng 3	373,5	1,9528019	373,5	1,8758726

Kết quả bảng 1 cho thấy, thời gian xử lý của hệ thống trên máy tính cá nhân (laptop core i3, 4 lõi, tốc độ xung nhịp 2,3 Ghz) là nhanh, cho thấy khả năng đề áp dụng trong thực tiễn. Quãng đường kết quả của lộ trình và thời gian xử lý của cả hai thuật toán khác nhau chưa nhiều do ba mô hình kho được xây dựng đơn giản. Tuy nhiên, kết quả thử nghiệm với nhiều loại kho và nhiệm vụ khác nhau cho thấy, hệ thống vận hành tốt, đưa ra kết quả nhanh và chính xác.

#### 4. Kết luận

Từ thuật toán tìm đường đi tối ưu của từng robot đơn lẻ, nhóm nghiên cứu áp dụng cho nhiều robot với cải tiến về mặt quy tắc tính chi phí nhỏ nhất cho toàn hệ robot bằng cách phân công nhiệm vụ, chiều di chuyển, thứ tự ưu tiên... để phù hợp với bài toán trong thực tế: kho hàng mật độ cao, nhiều robot cùng hoạt động, công ra tập trung tại số ít điểm, từ đó xây dựng được một hệ thống điều khiển hệ nhiều robot, có khả năng phối

hợp, tương tác trong quá trình thực hiện nhiệm vụ mà vẫn đảm bảo được trạng thái tối ưu của toàn hệ thống. Hệ thống được áp dụng thử nghiệm tại 3 kho hàng thực tế, với robot vật lý chạy thực tế và cập nhật lộ trình di chuyển theo thời gian thực trên giao diện chương trình máy tính cho kết quả xử lý tốt, với thời gian xử lý nhanh và quỹ đạo đã được kiểm chứng là đạt mức độ tối ưu vận hành. Tuy nhiên, nếu trên quy mô ứng dụng công nghiệp, cần phải tiếp tục thử nghiệm thêm các thuật toán khác trên các mô hình kho phức tạp hơn nữa để đáp ứng với các điều kiện thực tế. Ngoài ra, cần nghiên cứu thêm các quy tắc vận hành mang tính thống nhất vừa có thể áp dụng cho mọi sơ đồ kho hàng khác nhau, vừa đạt được yêu cầu về tối thiểu chi phí cho toàn hệ thống cũng như khả năng vận hành liên tục, không có sự cố, giảm thiểu tác động của con người, tạo nên một hệ thống kho hàng có tính tự động cao.

## LỜI CẢM ƠN

Các tác giả xin cảm ơn đề tài “Nghiên cứu, phát triển tổ hợp robot có tích hợp một số công nghệ chủ chốt của Cách mạng công nghiệp lần thứ tư, ứng dụng trong logistics”, mã số KC-4.0-34/19-25 đã hỗ trợ kinh phí, giúp đỡ thực hiện nghiên cứu này.

## TÀI LIỆU THAM KHẢO

- [1] E. Eade, T. Drummond (2007), “Monocular SLAM as a graph of coalesced observations”, *2007 IEEE 11<sup>th</sup> International Conference on Computer Vision*, pp.1-8, DOI: 10.1109/ICCV.2007.4409098.
- [2] F. Dellaert, M. Kaess (2006), “Square root SAM: Simultaneous localization and mapping via square root information smoothing”, *The International Journal of Robotics Research*, **25(12)**, pp.1181-1203, DOI: 10.1177/0278364906072768.
- [3] H. Strasdat, J. Montiel, A.J. Davison (2010), “Scale drift-aware large scale monocular SLAM”, *Robotics: Science and Systems*, **3(2)**, DOI: 10.7551/mitpress/9123.003.0014.
- [4] B. Williams, M. Cummins, J. Neira, et al. (2009), “A comparison of loop closing techniques in monocular SLAM”, *Robotics and Autonomous Systems*, **57(12)**, pp.1188-1197, DOI: 10.1016/j.robot.2009.06.010.
- [5] H.D. Whyte, T. Bailey (2006), “Simultaneous localization and mapping: Part I”, *IEEE Robotics & Automation Magazine*, **13(2)**, pp.99-110, DOI: 10.1109/MRA.2006.1638022.
- [6] B. Williams, M. Cummins, J. Neira, et al. (2008), “An image-to-map loop closing method for monocular SLAM”, *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp.2053-2059, DOI: 10.1109/IROS.2008.4650996.
- [7] M. Cummins, P. Newman (2008a), “FAB-MAP: Probabilistic localization and mapping in the space of appearance”, *The International Journal of Robotics Research*, **27(6)**, pp.647-665, DOI: 10.1177/0278364908090961.
- [8] M. Cummins, P. Newman (2008b), “Accelerated appearance-only SLAM”, *2008 IEEE International Conference on Robotics and Automation*, pp.1828-1833, DOI: 10.1109/ROBOT.2008.4543473.
- [9] S. Perera, A. Pasqual (2011), “Towards realtime handheld monoslam in dynamic environments”, *Advances in Visual Computing: 7<sup>th</sup> International Symposium*, pp.313-324, DOI: 10.1007/978-3-642-24028-7\_29.
- [10] R. Ozawa, Y. Takaoka, Y. Kida, et al. (2005), “Using visual odometry to create 3D maps for online footstep planning”, *2005 IEEE International Conference on Systems, Man and Cybernetics*, **3**, pp.2643-2648, DOI: 10.1109/ICSMC.2005.1571548.
- [11] D. Nistér (2005), “Preemptive RANSAC for live structure and motion estimation”, *Machine Vision and Applications*, **16(5)**, pp.321-329, DOI: 10.1109/ICCV.2003.1238341.
- [12] P. Michel, J. Chestnutt, S. Kagami, et al. (2007), “GPU-accelerated real-time 3D tracking for humanoid locomotion and stair climbing”, *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp.463-469, DOI: 10.1109/IROS.2007.4399104.
- [13] H.T. Tran, T.C. Vo, Q.N. Nguyen, et al. (2022), “A novel design of a smart interactive guiding robot for busy airports”, *International Journal on Smart Sensing and Intelligent Systems*, **15(1)**, DOI: 10.2478/ijssis-2022-0017.
- [14] Q. Liu, Y.G. Lu, C.X. Xie (2006), “Optimal genetic fuzzy obstacle avoidance controller of autonomous mobile robot based on ultrasonic sensors”, *2006 IEEE International Conference on Robotics and Biomimetics*, pp.125-129, DOI: 10.1109/ROBIO.2006.340327.
- [15] M. Algabri, H. Mathkour, H. Ramdane, et al. (2015), “Comparative study of soft computing techniques for mobile robot navigation in an unknown environment”, *Computers in Human Behavior*, **50**, pp.42-56, DOI: 10.1016/j.chb.2015.03.062.
- [16] M. Faisal, R. Hedjar, M.A. Sulaiman, et al. (2013), “Fuzzy logic navigation and obstacle avoidance by a mobile robot in an unknown dynamic environment”, *International Journal of Advanced Robotic Systems*, **10(1)**, DOI: 10.5772/54427.
- [17] H. Xiao, L. Liao, F. Zhou (2007), “Mobile robot path planning based on Q-ANN”, *2007 IEEE International Conference on Automation and Logistics*, pp.2650-2654, DOI: 10.1109/ICAL.2007.4339028.
- [18] C.T. Vo, N.A.Q. Nguyen, L.T.D. Tran, et al. (2021), “Fusion of inertial and magnetic sensors for autonomous vehicle navigation and freight in distinctive environment”, *International Conference on Engineering Research and Applications*, pp.431-439, DOI: 10.1007/978-3-030-92574-1.
- [19] S.X. Yang, T. Hu, X. Yuan, et al. (2003), “A neural network based torque controller for collision-free navigation of mobile robots”, *2003 IEEE International Conference on Robotics and Automation*, **1**, pp.13-18, DOI: 10.1109/ROBOT.2003.1241566.
- [20] A. Zhu, S.X. Yang (2007), “Neurofuzzy-based approach to mobile robot navigation in unknown environments”, *IEEE Transactions on Systems*, **37(4)**, pp.610-621, DOI: 10.1109/TSMCC.2007.897499.
- [21] S.U. Deshpande, S.S. Bhosale (2013), “Adaptive neuro-fuzzy inference system based robotic navigation”, *2013 IEEE International Conference on Computational Intelligence and Computing Research*, pp.1-4, DOI: 10.1109/ICCIC.2013.6724153.
- [22] M. Gao, J. Tian (2007), “Path planning for mobile robot based on improved simulated annealing artificial neural network”, *Third International Conference on Natural Computation*, **3**, pp.8-12, DOI: 10.1109/ICNC.2007.
- [23] A. Alyasin, E.I. Abbas, S.D. Hasan (2019), “An efficient optimal path finding for mobile robot based on Dijkstra method”, *2019 4<sup>th</sup> Scientific International Conference Najaf (SICN)*, pp.11-14, DOI: 10.1109/SICN47020.2019.9019345.
- [24] Q.N. Tang, V.T. Hoang, A.V. Nguyen, et al. (2023), “Development of an autonomous mobile robot system for hospital logistics in quarantine zones”, *The International Conference on Intelligent Systems & Networks*, pp.271-281, DOI: 10.1007/978-981-99-4725-6\_34.
- [25] N.K. Nguyen, V.T. Nguyen (2023), “Application of Dijkstra’s algorithm in mobile robot path-planning for items sorting function”, *Vietnam Electromechanical Magazine*, **302**, pp.78-83 (in Vietnamese).