NGHIÊN CỨU, THIẾT KẾ HỆ THỐNG ĐIỀU KHIỂN CHO ROBOT DI ĐỘNG CẤU TRÚC SKID-STEERING TRÊN HỆ ĐIỀU HÀNH ROS2

A STUDY AND DESIGN OF NAVIGATION SYSTEM FOR A SKID-STEERING MOBILE ROBOT BASED ON ROS2

Nguyễn Anh Tú, Vũ Văn Tài, Trần Tuấn Minh, Phan Thái Học

Trường Cơ khí - Ô tô, Trường Đại học Công nghiệp Hà Nội

Đến Tòa soạn ngày 24/3/2025, chấp nhận đăng ngày 15/4/2025

Tóm tắt: Bài báo này trình bày giải pháp thiết kế hệ thống điều khiển cho robot di động cấu trúc Skid-steering được triển khai trên hệ điều hành robot (ROS2). Hệ thống định vị cho robot sử dụng các cảm biến encoder, la bàn số và cảm biến quét laser để thu thập thông tin về môi trường làm việc. Quỹ đạo chuyển động của robot được thiết lập sử dụng thuật toán A* và quỹ đạo cục bộ được xây dựng sử dụng phương pháp Dynamic Window Based (DWB) cho bài toán tránh vật cản. Kết quả mô phỏng trên phần mềm Gazebo cho thấy robot thành công trong việc thiết lập quỹ đạo chuyển động, đồng thời tránh được các vật cản để di chuyển từ điểm từ điểm xuất phát đến điểm đích đặt trước.

Từ khóa: Robot di động, hệ điều hành robot, robot cấu trúc Skid-steering, thuật toán A*, phương pháp DWB.

Abstract: This paper presents a control system design for a Skid-steering mobile robot, implemented on the Robot Operating System (ROS2). The robot's localization system includes encoder sensors, a digital compass, and a laser scanner, which are used to collect environmental data. The system employs the A algorithm* for path planning based on a 2D grid map and the Dynamic Window Based (DWB) for obstacle avoidance. The simulation results in a Gazebo simulation environment show that the robot is successfully generate moving path and avoid obstacles to move to the goal points.

Keywords: Mobile robot, robot operation system, skid-steering robot, A* algorithm, DWB method.

1. GIỚI THIỆU

Trong bối cảnh cuộc Cách mạng công nghiệp 4.0, robot di động ngày càng đóng vai trò quan trọng và được ứng dụng trong nhiều lĩnh vực như logistics, cứu hộ và thám hiểm. Trong đó, robot di động dạng Skid-steering (SSMR) được ứng dụng bởi cho phép chịu tải lớn, khả năng cơ động cao trên các địa hình phức tạp [1]. Tuy nhiên, việc điều khiển chính xác loại robot này vẫn là thách thức lớn do đặc tính động học phi tuyến và hiện tượng trượt khi chuyển hướng [2].

Hệ điều hành robot (ROS) [3, 4] đã trở thành nền tảng phổ biến trong nghiên cứu và phát triển hệ thống điều khiển robot nhờ kiến trúc phân tán, khả năng tích hợp đa cảm biến và thư viện thuật toán phong phú. Bên cạnh đó, với sự hỗ trợ của các công cụ mô phỏng như Gazebo và RViz, ROS cho phép triển khai thiết kế, thử nghiệm nhanh chóng trước khi áp dụng vào thực tế. Nghiên cứu này trình bày một hệ thống điều khiển cho SSMR trên nền tảng ROS2 [5], bao gồm các đóng góp chính sau: Đề xuất mô hình điều khiển SSMR trên hệ điều hành ROS2; Hệ thống điều hướng có khả năng thiết kế quỹ đạo toàn cục sử dụng thuật toán A* [6] dựa trên bản đồ 2D và thiết

kế quỹ đạo cục bộ dựa trên phương Dynamic Window Based (DWB) [7] cho bài toán tránh vật cản [8].

Kết quả mô mô phỏng trên nền tảng Gazebo cho thấy robot có thiết lập quỹ đạo thành công trong môi trường làm việc được xác định trước, đồng thời có khả năng thiết lập lại quỹ đạo để tránh vật cản và di chuyển đến điểm đích mong muốn.

2. XÂY DỰNG HỆ THỐNG ĐIỀU KHIỂN TRÊN HỆ ĐIỀU HÀNH ROS

2.1. Mô hình động học của robot

Để khảo sát mô hình động học của SSMR, robot được xem như chuyển động trên mặt phẳng của hệ trục tọa độ vuông góc (Hình 1). Hệ trục tọa độ tương đối gắn trên robot có gốc tọa độ đặt tại trọng tâm (COM). Tọa độ của COM trong trong hệ trục tọa độ cố định là COM = (*X*, *Y*, *Z*) [9].

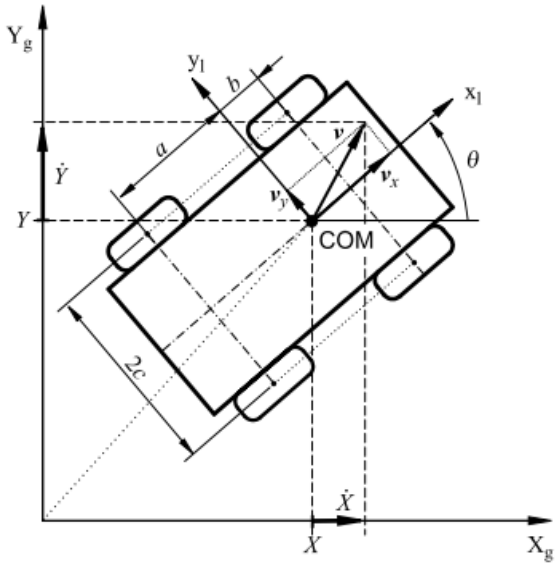
Giả sử robot di chuyển trên mặt phẳng với vận tốc tuyến tính được biểu thị hệ trục tọa độ tương đối là $v = \begin{bmatrix} v_x & v_y & 0 \end{bmatrix}^T$ và quay với vectơ vận tốc góc $\omega = \begin{bmatrix} 0 & 0 & \omega \end{bmatrix}^T$. Gọi $q = \begin{bmatrix} X & Y & \theta \end{bmatrix}^T$ là vectơ trạng thái mô tả trạng thái của robot (tức vị trí COM, *X* và *Y* và hướng θ của hệ trục tọa độ tương đối với với hệ trục tọa độ cố định), thì vectơ vận tốc của robot trong hệ tọa độ cố định được biểu diễn: $\dot{q} = \begin{bmatrix} \dot{X} & \dot{Y} & \dot{\theta} \end{bmatrix}^T$.

Từ hình vẽ có thể biểu diễn $\dot{X}$ và $\dot{Y}$ theo các vận tốc trên hệ trục tọa độ robot như sau:

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix} \quad (1)$$

Vì robot chuyển động trên mặt phẳng nên có thể biểu diễn $\dot{\theta} = \omega$.

Có thể thấy phương trình (1) không có ràng buộc nào đối với chuyển động của mặt phẳng SSMR, vì nó chỉ mô tả trạng thái động học. Do đó, cần phải phân tích mối quan hệ giữa vận tốc bánh xe và vận tốc thành phần trong hệ tọa độ robot. Giả sử bánh xe thứ *i* quay với vận tốc góc $\omega_i(t)$, trong đó *i* = 1, 2, …, 4. Các đại lượng v_L và v_R biểu thị tọa độ dọc của vận tốc bánh xe trái và phải, v_F và v_B lần lượt là thành phần vận tốc của bánh trước và bánh sau.



Hình 1. Sơ đồ biểu diễn các hệ trục tọa độ robot

Thực hiện các phép biến đổi ta có biểu thức mô tả mối quan hệ giữa vận tốc bánh xe và vận tốc của robot:

$$\begin{bmatrix} v_L \\ v_R \\ v_F \\ v_B \end{bmatrix} = \begin{bmatrix} 1 & -c \\ 1 & c \\ 0 & -x_{ICR}+b \\ 0 & -x_{ICR}-a \end{bmatrix} \begin{bmatrix} v_x \\ \omega \end{bmatrix} \quad (2)$$

Giả sử bán kính hiệu dụng của mỗi bánh xe là $r_i = r$, ta có:

$$\omega_\omega = \begin{bmatrix} \omega_L \\ \omega_R \end{bmatrix} = \frac{1}{r} \begin{bmatrix} v_L \\ v_R \end{bmatrix} \quad (3)$$

Với ω_L và ω_R lần lượt là các vận tốc góc của bánh xe bên trái và phải.

Kết hợp (2) và (3) ta có phương trình biểu diễn quan hệ đây giữa vận tốc góc bánh xe và vận tốc của robot:

$$\eta = \begin{bmatrix} v_x \\ \omega \end{bmatrix} = r \begin{bmatrix} \frac{\omega_L + \omega_R}{2} \\ \frac{-\omega_L + \omega_R}{2c} \end{bmatrix} \quad (4)$$

Trong đó η là thông số điều khiển đầu vào ở cấp độ động học.

Từ phương trình cuối cùng ta thấy, cặp vận tốc ω_L và ω_R có thể được coi là tín hiệu đầu vào động học điều khiển cũng như vận tốc v_x và ω. Tuy nhiên, độ chính xác của mối quan hệ (4) chủ yếu phụ thuộc vào sự trượt dọc. Ngoài ra, các tham số r và c có thể được xác định bằng thực nghiệm để đảm bảo sự tin cậy của việc xác định vận tốc robot đối với vận tốc góc của bánh xe.

Để hoàn thiện mô hình động học của SSMR, ràng buộc động học được đề xuất:

$$v_y + x_{ICR}\dot{\theta} = 0 \quad (5)$$

Phương trình (5) có thể được biểu diễn lại như sau:

$$\begin{bmatrix} -\sin\theta & \cos\theta & x_{ICR} \end{bmatrix} \begin{bmatrix} \dot{X} & \dot{Y} & \dot{\theta} \end{bmatrix} = A(q)\dot{q} = 0 \quad (6)$$

Vì vectơ vận tốc tổng quát $\dot{q}$ luôn nằm trong không gian rỗng của A, sử dụng ma trận biến đổi hệ trục tọa độ tổng quát, $S \in \mathbb{R}^{3x2}$, ta có thể viết:

$$\dot{q} = S(q)\eta \quad (7)$$

Trong đó:

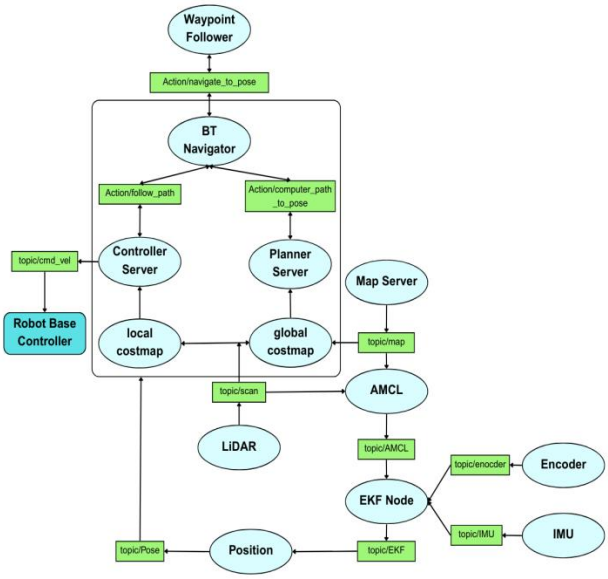
$$S^T(q)A^T(q) = 0 \quad (8)$$

$$S(q) = \begin{bmatrix} \cos\theta & x_{ICR}\sin\theta \\ \sin\theta & -x_{ICR}\cos\theta \\ 0 & 1 \end{bmatrix} \quad (9)$$

Vì $\dim(\eta) = 2 < \dim(q) = 3$, ta có phương trình (2.14) mô tả động học của robot

2.2. Sơ đồ hệ thống điều khiển

Để robot có đi đến điểm đích một cách chính xác thì giải pháp điều hướng đóng vai trò then chốt. Hệ thống điều hướng được thiết kế nhằm kiểm soát quá trình di chuyển của robot dựa trên thông tin về vị trí, hướng di chuyển và môi trường xung quanh tại mỗi thời điểm. Các dữ liệu cần thiết cho điều hướng được thu thập từ cảm biến encoder, la bàn số, và cảm biến quét laser.



Hình 2. Sơ đồ các node hệ thống điều hướng trên ROS2

Trong nghiên cứu này, giải pháp xây dựng hệ thống điều hướng cho robot dựa trên gói Navigation2 Stack của ROS2 nhằm giúp robot di chuyển tự động và an toàn trong môi trường làm việc, trong đó cấu trúc hệ thống được chia thành các node độc lập, mỗi node đảm nhận một chức năng cụ thể. Các node trao đổi thông tin với nhau thông qua các tin nhắn sử dụng các giao thức chuẩn đã được định nghĩa sẵn, tạo thành một hệ thống liên kết thống nhất và thuận lợi cho việc mở rộng.

Các thành phần trong kiến trúc ROS2 được

xây dựng bao gồm các node thuộc Navigation2 stack, cảm biến LiDAR 2D cho phát hiện vật cản, và một bản đồ tĩnh được sử dụng cho lập kế hoạch chuyển động. Sơ đồ kiến trúc của hệ thống được mô tả trong Hình 2.

Bảng 1. Chức năng các node trong hệ thống

STT	Tên node	Chức năng
1	BT Navigator	Nhận điểm đích từ node Waypoint follower và thực hiện điều phối quá trình điều hướng.
2	Planner Server	Sử dụng thuật toán A* để tìm và tạo đường đi từ vị trí hiện tại đến mục tiêu.
3	Controller Server	Sử dụng thuật toán DWB để điều khiển chuyển động của robot theo đường đi đã lên kế hoạch và tránh vật cản khi cần thiết.
4	Waypoint Follower	Nhận đầu vào điểm đích (tọa độ x, y, hướng) sau đó gửi điểm đích đến node BT Navigator Server để kích hoạt quy trình điều hướng.
5	Local costmap	Phát hiện vật cản trong phạm vi kiểm soát.
6	Global costmap	Xây dựng bản đồ tĩnh từ dữ liệu bản đồ được cung cấp trước.
7	Encoder	Đọc tín hiệu chuyển động quay của các bánh xe.
8	IMU	Đọc tín hiệu từ la bàn số để tính toán vận tốc dài và vận tốc góc của robot.
9	LiDAR	Đọc dữ liệu đám mây điểm của cảm biến laser đưa ra bản đồ cục bộ.
10	Map server	Cung cấp dữ liệu của bản đồ môi trường được cập nhật trước.
11	AMCL	So sánh thông tin giữa đám mây điểm hiện tại và bản đồ làm việc toàn cục để đưa ra được vị trí và hướng của robot.
12	EKF	Kết hợp các tín hiệu định vị từ encoder, la bàn số, bản đồ số để ước lượng về vị trí và góc của robot.
13	Position	Chứa thông tin về vị trí và góc của robot phục vụ quá trình điều khiển chuyển động.

Giải pháp định vị được thiết kế với node Encoder thực hiện việc tính toán vận tốc dài và vận tốc góc của robot. Node IMU có chức năng đọc dữ liệu của la bàn số và trả về dữ liệu đầu ra là góc và vận tốc góc của robot. Thông tin của bản đồ của robot được xác định qua các đám mây điểm từ cảm biến quét laser, dữ liệu này được node AMCL đối sánh với bản đồ của môi trường làm việc được nhập trước vào hệ thống để tìm ra vị trí của robot trên bản đồ. Các thông tin định vị từ ba loại cảm biến trên được đưa vào vào node EKF để xác định vị trí và hướng của robot tại mỗi thời điểm phục vụ quá trình điều khiển chuyển động của robot. Node BT Navigator Server hoạt động như một trung tâm điều phối toàn bộ tiến trình điều hướng. Khi node Waypoint Follower gửi Action/navigate_to_pose đến node BT Navigator thì node sẽ có nhiệm vụ điều phối đầu tiên thực hiện gọi node Planner Server. Node Planner Server sẽ sử dụng dữ liệu đầu vào là computer_path_to_pose, Pose và global costmap để tính toán và tạo ra đường dẫn từ vị trí hiện tại đến đích. Dữ liệu đường dẫn sẽ được chuyển tiếp đến node Controller Server thông qua Action/follow_path để node có thể tính toán và điều khiển robot di chuyển theo đường đi

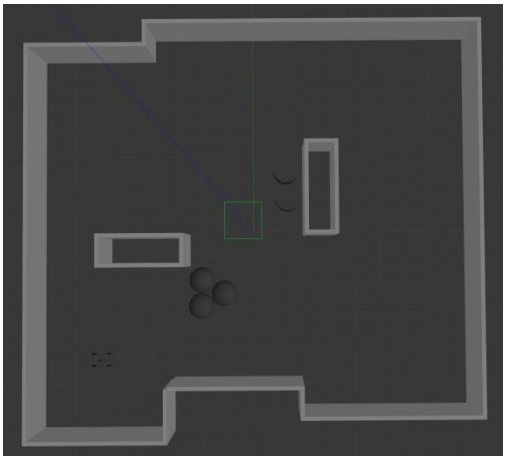
đã lập kế hoạch. Trong quá trình thực thi, node Controller Server sẽ sử dụng thông tin từ local costmap để tránh vật cản động và tối ưu hóa chuyển động. Trong hệ thống phát triển trên ROS2, các node hoạt động với nhiệm vụ và chức năng độc lập, do đó khi cần thay đổi hoặc nâng cấp chỉ cần thực hiện trên từng node cụ thể mà không ảnh hưởng đến toàn bộ hệ thống.

3. XÂY DỰNG MÔ HÌNH MÔ PHỎNG, THỬ NGHIỆM VÀ ĐÁNH GIÁ HỆ THỐNG

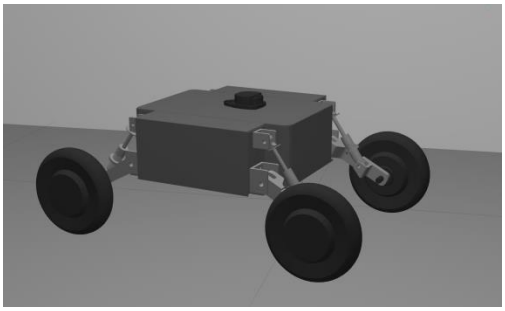
3.1. Xây dựng mô hình mô phỏng

Để đánh giá giải pháp đề xuất, mô hình robot và môi trường làm việc được xây dựng trên nền tảng phần mềm Gazebo và mô phỏng trên phần mềm Rviz. Môi trường hoạt động của robot được bao quanh bởi các tường bao, các khu vực hình chữ nhật tượng trưng cho các vật cản gắn liền với môi trường làm việc, đây là khu vực robot không được phép di chuyển vào (Hình 3). Mô hình robot dạng SSRM được thiết kế như Hình 4 với thiết kế bao gồm khung xe, bốn động cơ dẫn động cho bốn bánh xe. Các cảm biến gồm có encoder, la bàn số và cảm biến quét laser. Thông số của cảm biến được mô tả như trong Bảng 2.

Để thử nghiệm khả năng thiết lập quỹ đạo và tránh vật cản của robot, trước tiên công cụ Google Cartographer được sử dụng để thu thập thông tin về môi trường hoạt động và thiết lập bản đồ làm việc. Bản đồ sau đó được tích hợp vào các khối cost_map và move_base để phục vụ cho quá trình điều hướng, đồng thời được mô phỏng trên Rviz. Bản đồ sau khi xử lý trong cost_map, dải màu sẽ có giá trị từ 0 đến 255, trong đó các vùng màu đen biểu thị đường bao của vật cản, màu trắng là không gian cho phép robot chuyển động. Đường bao màu tím xung quanh các vật cản được định nghĩa để tạo thêm vùng an toàn nhằm đảm bảo robot không va chạm trong quá trình di chuyển.



Hình 3. Môi trường giả lập



Hình 4. Mô hình 3D của robot SSMR

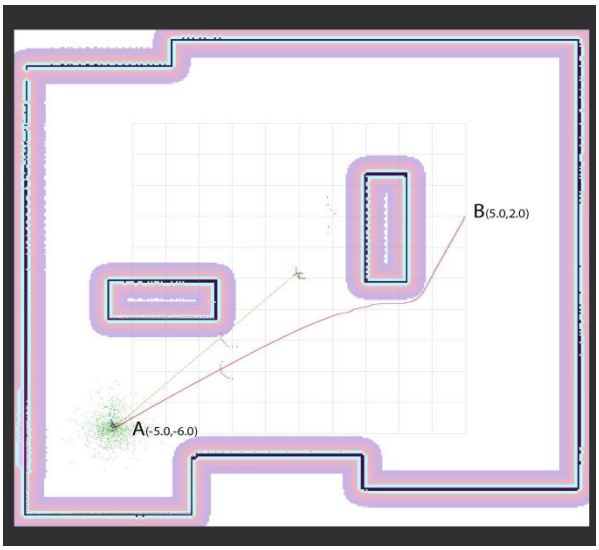
Bảng 2. Thông số cảm biến

Cảm biến Laser	- Khoảng cách đo: 0,15-12 m - Góc quét: 360^o - Tần số làm việc: 5,5 Hz
La bàn số	- Trả về thông tin góc quay của đối tượng với độ phân giải 0,015^o - Tần số làm việc: 200 Hz
Encoder tương đối	- Độ phân giải: 500 xung/vòng - Tần số đáp ứng: 100 kHz

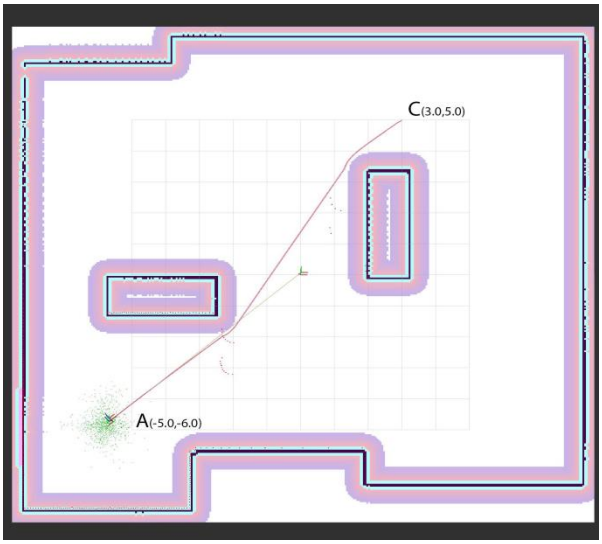
Hệ thống định vị của robot sử dụng kết hợp cả AMCL và EKF, trong đó EKF đóng vai trò như là bộ lọc trung tâm, tổng hợp hợp dữ liệu cảm biến encoder và la bàn số để ước tính liên tục vị trí của robot. AMCL cung cấp vị trí tuyệt đối để hiệu chỉnh sai số tích lũy của EKF.

3.2. Kết quả thử nghiệm và bàn luận

Robot được được thử nghiệm chuyển động theo hai trường hợp: Xuất phát từ điểm A có tọa độ (–5,0; –6,0) chuyển động đến điểm B có tọa độ (5,0; 2,0); xuất phát từ điểm A chuyển động đến điểm C có tọa độ (3,0; 5,0). Vận tốc dài và vận tốc góc lớn nhất được thiết lập lần lượt là 0,26 m/s và 1 rad/s.

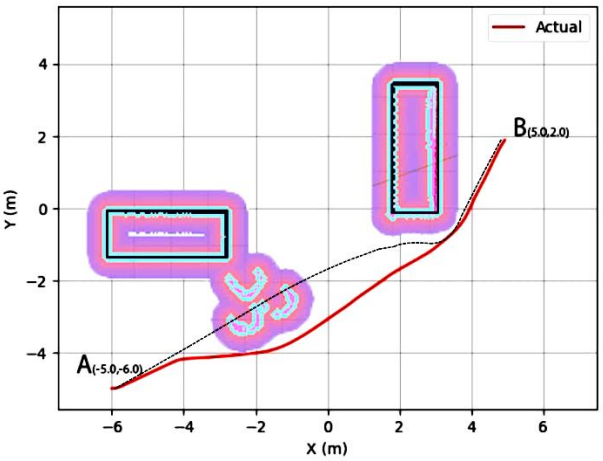


Hình 5. Quỹ đạo chuyển động của robot đến điểm B khi chưa xuất hiện vật cản

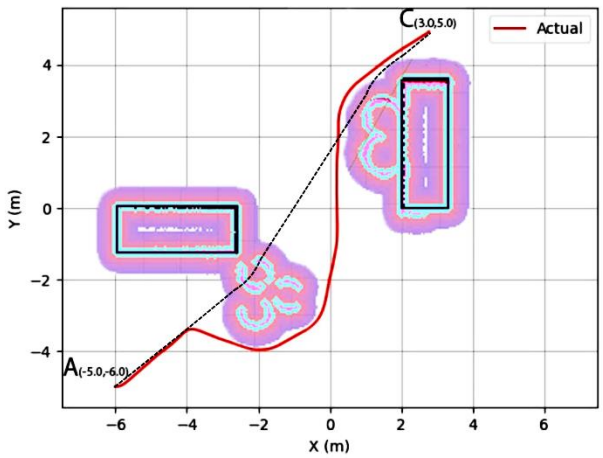


Hình 6. Quỹ đạo chuyển động của robot đến điểm C khi chưa xuất hiện vật cản

Quỹ đạo chuyển động của robot tạo ra bởi thuật toán A* trong hai trường hợp được biểu diễn như trên Hình 5 và 6. Sau đó, để đánh giá khả năng tránh vật cản của robot với thuật toán DWB, các khối vật cản tròn được bổ sung vào môi trường làm việc (những vật cản này không được định nghĩa trên bản đồ của môi trường làm việc). Hình 7 và 8 mô tả quỹ đạo chuyển động của robot khi chưa có vật cản và khi có vật cản. Kết quả mô phỏng cho thấy, khi robot chuyển động đến khu vực có vật cản, quỹ đạo mới được thiết lập để tránh các vật cản, ở các khu vực khác robot vẫn chuyển động bám theo quỹ đạo đã được thiết lập theo thuật toán A* để đi đến điểm đích.



Hình 7. Quỹ đạo chuyển động của robot đến điểm B khi xuất hiện vật cản



Hình 8. Quỹ đạo chuyển động của robot đến điểm C khi xuất hiện vật cản

4. KẾT LUẬN

Bài báo đã trình bày một giải pháp thiết kế hệ

thống điều khiển cho SSMR dựa trên hệ điều hành robot ROS2. Hệ thống định vị được thiết kế sử dụng đa cảm biến, trong đó bộ lọc EKF được sử dụng để tổng hợp dữ liệu định vị. Bài toán thiết kế quỹ đạo chuyển động cho robot sử dụng thuật toán A^* kết hợp với thuật toán DWA để tránh vật cản. Kết quả mô phỏng cho thấy robot đã thiết lập thành công quỹ đạo để robot chuyển động đến điểm đích đặt trước trong hai trường hợp chuyển động theo bản đồ thiết lập trước và trường hợp xuất hiện vật cản trong môi trường làm việc. Bên cạnh đó, giải pháp thiết kế hệ thống trên ROS2 cho phép người dùng có thể linh hoạt trong việc điều chỉnh, cấu trúc lại theo từng ứng dụng cụ thể.

TÀI LIỆU THAM KHẢO

[1] R. Khan, F. M. Malik, A. Raza, and N. Mazhar, "Comprehensive study of skid-steer wheeled mobile robots: Development and challenges," Ind. Robot: Int. J. Robot. Res. Appl., vol. 47, no. 5, pp. 623–638, (2020), doi: 10.1108/IR-04-2020-0082.

[2] J. Pentzer, S. Brennan, and K. Reichard, "The use of unicycle robot control strategies for skid-steer robots through the ICR kinematic mapping," 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Chicago, IL, USA, pp. 3201-3206, (2014), doi: 10.1109/IROS.2014.6942979.

[3] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A. Y. Ng., ROS: an open-source Robot Operating System. In ICRA Workshop on Open Source Software, (2009).

[4] Y. Pyo, H. Cho, R. W. Jung, T. Lim. ROS Robot Programming. Republic of Korea, ROBOTIS Co., Ltd, (2017).

[5] T. Duckett, S. Marsland, J. Shapiro. Learning globally consistent maps by relaxation. Proceeding of the IEEE International Conference on Robotics and Automation (ICRA), (2000).

[6] W. Zeng and R. L. Church, "Finding shortest paths on real road networks: the case for A*," International Journal of Geographical Information Science, vol. 23, no. 4, pp. 531–543, (2009). doi: 10.1080/13658810801949850.

[7] E.J. Molinos, Á. Llamazares, and M. Ocaña, "Dynamic window based approaches for avoiding obstacles in moving," Robotics and Autonomous Systems, vol. 118, pp. 112–130, (2019). doi: 10.1016/j.robot.2019.05.003.

[8] M. Seder and I. Petrović, "Dynamic window based approach to mobile robot motion control in the presence of moving obstacles," in 2007 IEEE International Conference on Robotics and Automation, Rome, Italy, pp. 1986-1991, (2007). doi: 10.1109/ROBOT.2007.363646.

[9] [9]. K. Kozłowski and D. Pazderski, "Modeling and control of a 4-wheel skid-steering mobile robot," Int. J. Appl. Math. Comput. Sci., Vol. 14, No. 4, 477- 496, (2004).

Thông tin liên hệ: **Nguyễn Anh Tú**

Điện thoại: 0904378033 - Email: tuna@haui.edu.vn

Trường Cơ khí - Ô tô, Trường Đại học Công nghiệp Hà Nội.