

DOI:10.22144/ctu.jvn.2023.030

## MÔ PHỎNG KHẢO SÁT CHẤT LƯỢNG CỦA HỆ THỐNG TÌM ĐƯỜNG VÀ BÁM ĐƯỜNG CHO XE TỰ HÀNH LÁI BẰNG NGUYÊN LÝ ACKERMANN TRÊN ROS VÀ GAZEBO

Trần Lê Trung Chánh\*, Ngô Quang Hiếu, Nguyễn Tùng Khánh, Mai Chiêu Linh, Trần Phước Lộc và Lê Văn Lê

Trường Bách khoa, Trường Đại học Cần Thơ

\*Người chịu trách nhiệm về bài viết: Trần Lê Trung Chánh (email: tlchanh@ctu.edu.vn)

### Thông tin chung:

Ngày nhận bài: 03/11/2022

Ngày nhận bài sửa: 18/11/2022

Ngày duyệt đăng: 02/12/2022

### Title:

Simulation and examination of Ackermann-driven Autonomous Mobile Robot with path-planning and path-following on ROS and Gazebo

### Từ khóa:

Robot tự hành, Robot di động, ROS, Gazebo

### Keywords:

Mobile robot, AMR, ROS, Gazebo

### ABSTRACT

This paper is aimed to study autonomous mobile robots according to the Ackermann principle. The robot model was built to simulate the localization, path-finding, and path-following sub-systems based on Robot Operating System (ROS) and Gazebo. The simulation method is based on the Behavior Tree structure and nodes in ROS 2, which navigated the robot moving to fixed position. Simulation results show that the proposed solution is better than the normal and multi-directional driving principle on slippery and rough terrain. The test results indicated that the robot successfully reached the target, where distance deviation is 96% for greater than or equal to 1 m and 79% for less than 1 m. This study effectively enables practical modeling regarding positioning by GPS.

### TÓM TẮT

Bài báo này được thực hiện nhằm nghiên cứu xe tự hành lái theo nguyên lý Ackermann. Mô hình xe được xây dựng để mô phỏng hệ thống định vị, tìm đường, bám đường dựa trên Gazebo và Robot Operating System (ROS). Mô phỏng thực hiện dựa trên cấu trúc Cây hành vi và các node được thiết lập trong ROS 2 nhằm đưa xe đến đúng vị trí theo yêu cầu. Kết quả mô phỏng cho thấy giải pháp đề xuất đã đáp ứng tốt hơn so với nguyên lý lái hai bánh chủ động và lái đa hướng khi thực hiện trên địa hình dễ bị trượt và gồ ghề. Kết quả khảo sát cho thấy robot di chuyển đến mục tiêu với độ lệch khoảng cách lớn hơn hoặc bằng 1 m đạt 96% và nhỏ hơn 1 m đạt 79%. Đồng thời, nghiên cứu này có thể làm tiền đề để triển khai mô hình thực tế trên cơ sở định vị bằng hệ thống GPS.

## 1. GIỚI THIỆU

Xe tự hành (Autonomous Mobile Robot – AMR) là nhóm robot di động có khả năng định vị và di chuyển một cách chuẩn xác trong môi trường hoạt động. Vấn đề này gọi là điều hướng robot (robot navigation). Nhìn chung, điều hướng robot có thể chia thành các bài toán con:

- Định vị: xác định vị trí robot trong môi trường làm việc. Robot dựa vào thông tin từ các cảm biến khác nhau để tự đánh giá vị trí hiện tại của bản thân và các vật cản.

- Tìm đường: lập kế hoạch đường đi ngắn nhất từ vị trí hiện tại đến vị trí mong muốn.

– Bám đường: điều khiển các cơ cấu chấp hành bám theo đường đi đã hoạch định. Nếu vật cản trong môi trường làm việc có thể thay đổi linh hoạt, thuật toán điều khiển bám đường có thể cần phải xử lý tránh vật cản.

Robot tự hành dạng xe thường có ba bậc tự do, được thể hiện qua các biến miêu tả vị trí và góc xoay của robot trong không gian làm việc. Tùy vào số tác động điều khiển, có thể chia nguyên lý lái robot thành hai kiểu: lái thường (hệ thiếu tác động điều khiển – underactuated) và lái đa hướng (hệ đủ tác động điều khiển – fully actuated).

Trong hai kiểu lái trên, xe lái đa hướng (Hiệp và ctv., 2020) (Hiếu và ctv., 2022) có khả năng di chuyển linh hoạt nhất, có tiềm năng ứng dụng cho nhiều môi trường hoạt động. Tuy nhiên, do đặc trưng về cơ học của nguyên lý lái đa hướng, sự chênh lệch về chất lượng điều khiển của từng cơ cấu chấp hành cộng với tỷ suất trượt có ảnh hưởng nhiều đến tính toán động học robot (Chí & Vân, 2019). Vì vậy, hiện tại, hầu hết ứng dụng xe đa hướng chỉ giới hạn ở địa hình bằng phẳng, ít trượt. Ngoài ra, việc phóng to thiết kế cũng gặp phải những vấn đề như: tăng chi phí chế tạo do có nhiều chi tiết phức tạp chưa được tiêu chuẩn hóa (như bánh xe đa hướng); chưa có nghiên cứu tổng quát được công bố về phương pháp thiết kế bộ khung động lực học cho xe đa hướng kích cỡ lớn.

Trong khi đó, am hiểu về mặt cơ học đối với xe lái thường đã phổ biến rộng rãi. Có hai nguyên lý lái được sử dụng nhiều: lái hai bánh chủ động và lái theo nguyên lý Ackermann.

Lái hai bánh chủ động – còn được gọi là lái “vi sai” (differential driven) – là điều khiển hướng dựa trên sự khác biệt tốc độ dẫn động hai bánh hoặc hai hàng bánh hai bên. Với nguyên lý lái này, xe có thể bẻ cua gắt và xoay được tại chỗ, giúp dễ tránh vật cản. Vì ưu điểm trên, nguyên lý lái hai bánh chủ động được sử dụng phổ biến trong nghiên cứu, với robot kích thước nhỏ, môi trường trong nhà (Hường, 2022; Štefek et al., 2021). Nhược điểm nói chung vẫn là khó điều khiển chính xác bán kính quay vòng do ảnh hưởng tương quan giữa nhiều tác động điều khiển như trường hợp lái đa hướng (Štefek và ctv., 2021) (Qin và ctv., 2021).

Ngược lại, các xe thiết kế lái theo nguyên lý Ackermann quay vòng bằng cách điều chỉnh trục các bánh dẫn hướng. Đường kéo dài các trục bánh xe đồng quy tại một điểm (tâm đường vòng) để giúp bánh không bị trượt ngang (trượt theo phương dọc trục). Do đó, nguyên lý lái Ackermann có ưu điểm

bảo vệ lốp xe và có tác động điều khiển bán kính quay vòng độc lập với tốc độ dài; là lựa chọn ưu tiên cho các robot hoạt động nhiều trên mặt đường (Elverum, 2022), hoặc cần di chuyển tốc độ cao (Lee, 2022). Chế tạo robot lái kiểu Ackermann cũng dần trở nên dễ dàng hơn vì thị trường hiện nay cung cấp các chi tiết cơ khí ô tô đã chuẩn hóa (các chi tiết trong bánh xe, bộ truyền vi sai, thước lái,...) với giá thành phù hợp.

Vấn đề lớn nhất đối với thiết kế dạng này là thường bị giới hạn góc lái, gây ra khó khăn cho việc giải quyết bài toán hoạch định chuyển động và tránh vật cản. Để giải quyết cho vấn đề này, nhiều nghiên cứu đã được công bố. Cụ thể, Dubins (1957) đã đề xuất cách tính biên dạng đường đi ngắn nhất cho xe có giới hạn góc lái và chỉ chạy tới. Tiếp đó, Reeds and Shepp (1990) mở rộng giới hạn bài toán cho các trường hợp xe chạy lùi. Từ đó đến nay, trong cộng đồng nghiên cứu đã có nỗ lực kết hợp các phương pháp trên với những giải thuật tránh vật cản. Điển hình là Dolgov et al. (2010), đã giới thiệu thuật toán tìm đường “Hybrid A\*”. Thuật toán được áp dụng trong cuộc thi DARPA Urban Challenge 2007 và đã phổ biến rộng rãi cho ứng dụng các kiểu robot tự hành có góc lái giới hạn.

Như vậy, nguyên lý lái kiểu Ackermann hội tụ đủ điều kiện để làm phương án thiết kế chi phí vừa phải, thời gian chế tạo ngắn, có thể ứng dụng thực tế. Bài báo này là một nỗ lực nhằm kết nối giữa nghiên cứu và ứng dụng robot tự hành thông qua việc thực hiện mô phỏng khảo sát chất lượng của hệ thống tìm đường và bám đường cho robot lái theo nguyên lý như đã nêu.

## 2. PHƯƠNG PHÁP NGHIÊN CỨU

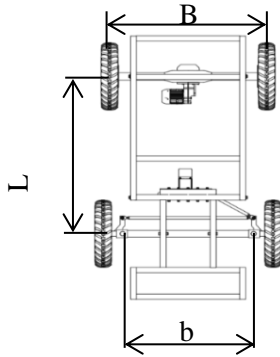
### 2.1. Phần cơ khí mô phỏng

Phần cơ khí mô phỏng robot được thiết kế dựa trên xe của nhóm luận văn (Tấn & Phúc, 2021). Các thông số quan trọng của xe được mô tả ở Hình 1 và Bảng 1. Thông số đầy đủ của xe, bao gồm kích thước, khối lượng, mô hình va chạm được mô tả trong (Khánh, 2022).

Theo nguyên lý lái Ackermann, góc lái của bánh trong lớn hơn góc lái bánh ngoài. Giới hạn bán kính lái của xe được tính như sau:

$$R_{min} = \frac{b}{2} + \frac{L}{\tan \alpha_{max}} \quad (1)$$

Thế  $b$ ,  $L$ ,  $\alpha_{max}$  vào, được giá trị  $R_{min}$  là 1591,5 mm.



Hình 1. Các kích thước quan trọng

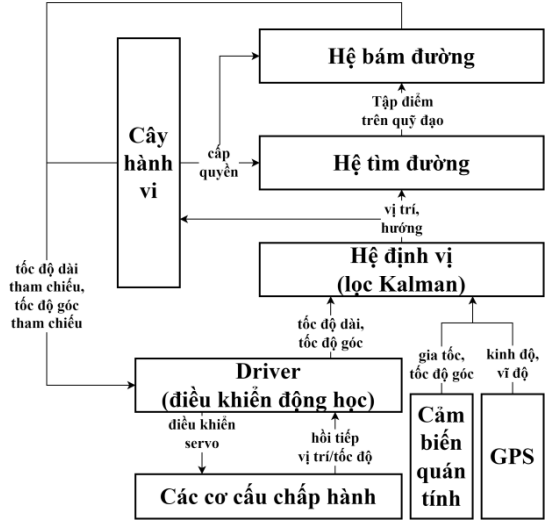
Bảng 1. Các thông số của xe

Thông số	Kí hiệu	Giá trị
Khoảng cách giữa hai khớp lái (pivot distance)	b	803 mm
Khoảng cách giữa hai bánh sau (wheel track)	B	1066 mm
Khoảng cách giữa cầu trước và cầu sau (wheel base)	L	1190 mm
Bán kính bánh xe (wheel radius)	$r_{bánh}$	250 mm
Góc lái tối đa (max steering angle)	$\alpha_{max}$	45°
Tốc độ xoay khớp lái (steering speed)	$\omega_{lái}$	90 °/s
Tốc độ tối đa	$v_{max}$	1 m/s
Gia tốc tối đa	$a_{max}$	1 m/s

2.2. Hệ thống phần mềm

Hệ thống phần mềm của robot được xây dựng trên Robot Operating System 2 (ROS 2), và vận hành thông qua các chương trình con chạy song song với nhau, gọi là node. Thông tin về cách vận hành của ROS 2 có thể được tham khảo trong Open Robotics (2022) và Macenski et al. (2022). Sơ đồ hệ thống được trình bày sơ lược trên Hình 2. Trong đó, hệ tìm đường được chạy trên node planner\_server, sử dụng thuật toán Hybrid A\* (Dolgov và ctv., 2010). Hệ bám đường chạy trên node controller\_server, với thuật toán bám đường được sử dụng là Pure Pursuit (Coulter, 1992). Hệ định vị là một tập hợp các node chạy Bộ lọc Kalman Mở rộng (Extended Kalman Filter – EKF): local\_ekf, global\_ekf và navsat. Dữ liệu cảm biến đầu vào bao gồm gia tốc, góc xoay đo được từ một cảm biến quán tính (bao gồm con quay hồi chuyển 3 trục và cảm biến gia tốc 3 trục), vận tốc dài, vận tốc góc từ động học thuận của bộ driver mô phỏng, và dữ liệu từ GPS. Thiết kế của các EKF được tham khảo ở (Moore & Stouch, 2014; Moore, 2016a, 2016b).

Thông số và nhiều của các cảm biến mô phỏng được cho ở Bảng 5. Thông số cài đặt thiết yếu của các node trên lần lượt được ghi lại ở các Bảng 2, Bảng 3 và Bảng 4. Thông số đầy đủ của hệ thống phần mềm có thể được tra cứu ở (Khánh, 2022).



Hình 2. Sơ đồ truyền dữ liệu giữa các thành phần trong hệ thống phần mềm

Bảng 2. Thông số hệ tìm đường

Thông số	Giá trị
Tần số lấy mẫu	20 Hz
Thuật toán	Hybrid A*
Phương pháp tìm biên dạng đường đi	Reed – Shepp
Bán kính cong tối thiểu cho phép	1,6 m
Hệ số ức chế đối với lựa chọn chạy lùi	2

Bảng 3. Thông số hệ bám

Thông số	Giá trị
Tần số lấy mẫu	20 Hz
Thuật toán	Pure Pursuit
Khoảng cách đến điểm cần bám (look ahead distance)	1 m
Tốc độ dài	0,5 m/s
Sai số khoảng cách cho phép	0,4 m
Sai số góc hướng đầu xe cho phép	4.58°

Bảng 4. Thông số hệ định vị (các bộ lọc EKF)

Thông số	Giá trị
Tần số lấy mẫu	30 Hz
Phương sai nhiễu vị trí	[0,001 0,001 0,001]
Phương sai nhiễu vận tốc	[0,5 0,5 0,1]
Phương sai nhiễu gia tốc	[0,3 0,3 0,3]
Phương sai nhiễu góc xoay	[0,3 0,3 0,1]
Phương sai nhiễu vận tốc góc	[0,3 0,3 0,3]

Ghi chú: đơn vị của dữ liệu theo tiêu chuẩn REP-103 (Foote & Purvis, 2010)

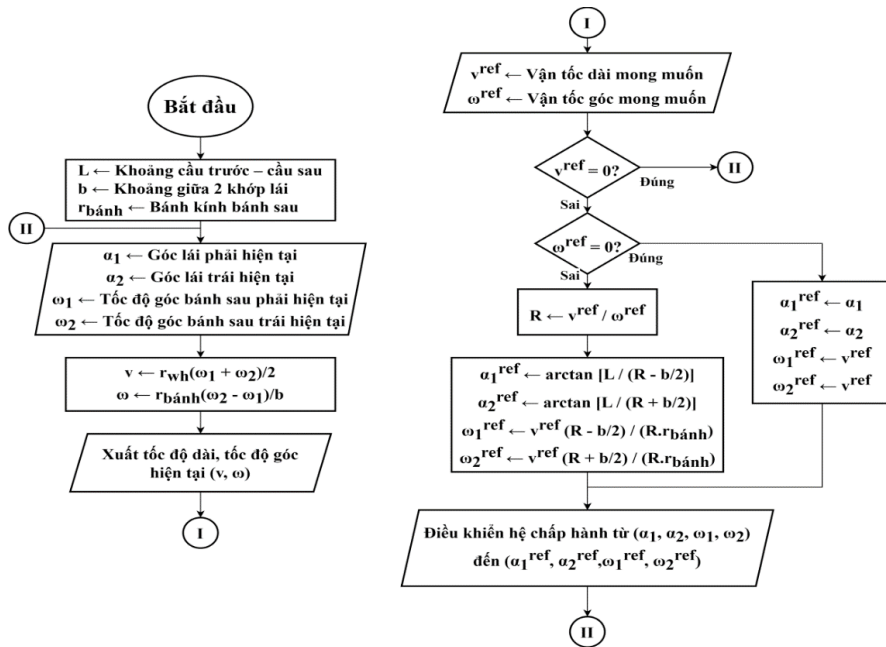
**Bảng 5. Thông số các cảm biến mô phỏng**

Cảm biến	Thông số	Giá trị
Con quay hồi chuyển	Tần số lấy mẫu	100 Hz
	Trung bình nhiễu	0 rad/s
	Độ lệch chuẩn nhiễu	0,0002 rad/s
	Trung bình độ lệch không	$7,5 \times 10^{-6}$ rad/s
	Độ lệch chuẩn của độ lệch không	$0,8 \times 10^{-6}$ rad/s
Cảm biến gia tốc	Tần số lấy mẫu	100 Hz
	Trung bình nhiễu	0 m/s <sup>2</sup>
	Độ lệch chuẩn nhiễu	0,017 m/s <sup>2</sup>
	Trung bình độ lệch không	0,1 m/s <sup>2</sup>
GPS	Độ lệch chuẩn của độ lệch không	0,001 m/s <sup>2</sup>
	Tần số lấy mẫu	1 Hz
	Độ lệch không của nhiễu vị trí	0,001 m
	Độ lệch chuẩn của nhiễu vị trí	0,05 m
	Độ lệch không của nhiễu vận tốc	0,001 m/s
	Độ lệch chuẩn của nhiễu vận tốc	0,5 m/s

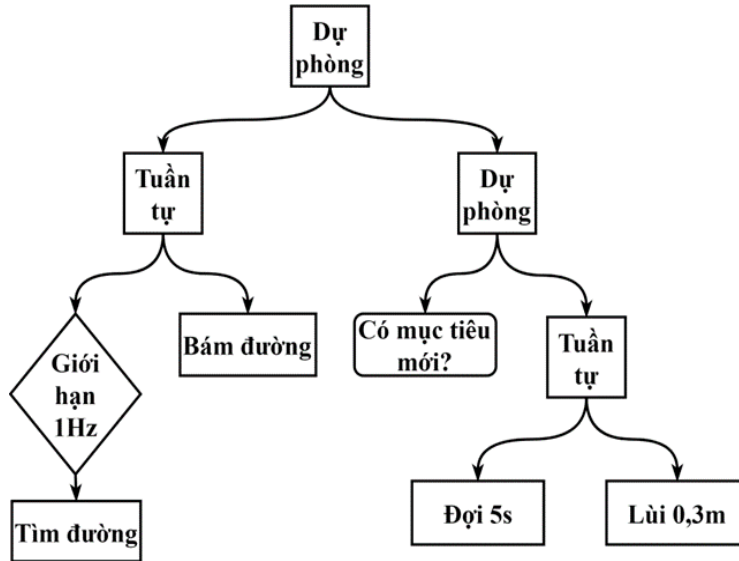
Thông thường, trong thiết kế xe lái theo nguyên lý Ackermann, hai bánh sau được dẫn động bằng một động cơ thông qua bộ truyền động vi sai, còn hai khớp lái của bánh trước cũng được điều khiển bằng một khâu dẫn thông qua một cơ cấu phẳng toàn khớp thấp. Trong nghiên cứu này, để đơn giản mô phỏng cơ học, mô hình xe được điều khiển bởi 4 cơ cấu chấp hành độc lập: 2 servo

điều tốc 2 bánh sau và 2 servo điều khiển vị trí 2 khớp lái. Lưu đồ bộ điều khiển động học được mô tả trên Hình 3.

Hệ quản lý tác vụ sử dụng Cây hành vi (Behavior Tree) được mô tả trong Hình 4. Về cơ bản, các tác vụ đều là nút lá của cây hành vi, và chỉ chạy khi được cây hành vi cấp quyền (ticking). Chi tiết về hoạt động của Cây hành vi và ưu điểm so với các phương pháp truyền thống hơn được mô tả trong (Colledanchise & Ögren, 2017).



**Hình 3. Lưu đồ chương trình điều khiển động học xe**



Hình 4. Cây hành vi

Logic hoạt động của Cây hành vi trong Hình 4 có thể được miêu tả sơ lược như sau:

- Tác vụ tìm đường và bám đường được cấp quyền hoạt động và được kiểm tra tuần tự với tần số 1 Hz.

- Khi phát hiện lỗi – do không tính được đường đi đến mục tiêu hoặc bám đường vượt sai số cho phép – hệ thống liên tục kiểm tra xem có lệnh di chuyển đến mục tiêu mới không, nếu không thì thực hiện lần lượt 2 tác vụ “phục hồi” (recovery): đợi trong 5 giây, sau đó lùi 0,3 m.

- Nếu không còn lỗi hoặc có lệnh di chuyển đến mục tiêu mới thì ngưng tác vụ “phục hồi”.

**2.3. Mô phỏng**

Phần mềm chạy mô phỏng cơ học là Gazebo 11. Phần cứng chạy mô phỏng là máy tính cá nhân với CPU Intel Core i5-8265U 1,6 MHz, RAM 4GB, card đồ họa Intel UHD Graphics 620 và NVIDIA GeForce MX130.

Trong quá trình mô phỏng, 64 thí nghiệm được tiến hành với số lần lặp lại mỗi thí nghiệm là 3. Trong mỗi phép thử, xe bắt đầu từ trạng thái đứng yên. Gọi vị trí và hướng của xe so với hệ quy chiếu cố định lúc này là  $x_0, y_0$  và  $\theta_0$ . Vị trí và góc hướng đầu xe của mục tiêu cần di chuyển tới  $x_{mt}, y_{mt}$  và  $\theta_{mt}$  được xác định như sau:

$$x_{mt} = x_0 + d_i \cos(\theta_0 + \varphi_i) \quad (2)$$

$$y_{mt} = y_0 + d_i \sin(\theta_0 + \varphi_i) \quad (3)$$

$$\theta_{mt} = \theta_0 + \Delta\theta_i \quad (4)$$

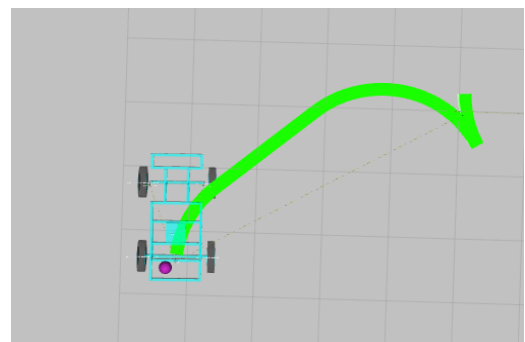
Trong đó,  $d_i, \varphi_i$  và  $\Delta\theta_i$  là tham số của thí nghiệm thứ  $i$ . Mỗi tham số mang một trong các giá trị cho ở Bảng 6.

**Bảng 6. Tập giá trị của từng tham số thí nghiệm**

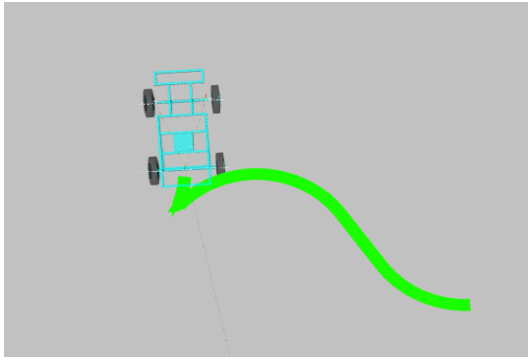
Tham số	Tập giá trị
$d$	{2,5 m ; 5 m}
$\varphi$	{0°, -60°, -120°, -180°}
$\Delta\theta$	{ $k \times 45^\circ \mid k \in \{0, \dots, 7\}$ }

**3. KẾT QUẢ VÀ THẢO LUẬN**

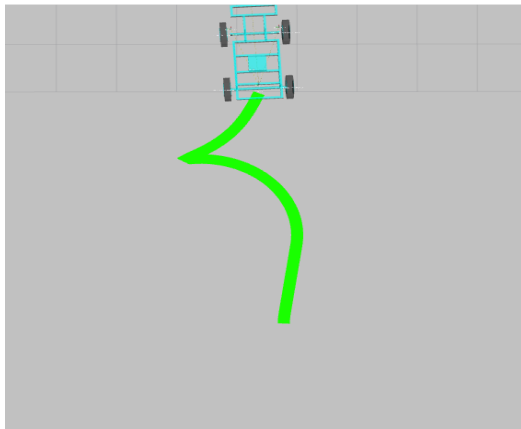
Video mô phỏng, thứ tự, tham số từng thí nghiệm và kết quả được ghi lại trong (Khánh, 2022). Trong đó, số liệu kết quả bao gồm thời gian để đến mục tiêu  $\Delta t_i$ , sai số khoảng cách và góc hướng đầu xe khi đến mục tiêu  $\varepsilon_{d_i}, \varepsilon_{\theta_i}$ .



Hình 5. Đường đi hoạch định ở thí nghiệm 45 ( $d_{45} = 5 \text{ m}$ ;  $\varphi_{45} = -60^\circ$ ;  $\Delta\theta_{45} = 180^\circ$ )



**Hình 6. Đường đi hoạch định ở thí nghiệm 55**  
( $d_{55} = 5 \text{ m}$ ;  $\phi_{55} = -120^\circ$ ;  $\Delta\theta_{55} = 270^\circ$ )



**Hình 7. Đường đi hoạch định ở thí nghiệm 60**  
( $d_{60} = 5 \text{ m}$ ;  $\phi_{60} = -180^\circ$ ;  $\Delta\theta_{60} = 135^\circ$ )

Hình 5, Hình 6 và Hình 7 mô tả một số biên dạng đường đi hoạch định.

Trong 192 lần thử với độ lệch khoảng cách  $\varepsilon_d$  trung bình là 0,7 m và lớn nhất là 1,9 m; độ lệch góc hướng đầu xe  $\varepsilon_\theta$  trung bình là  $4^\circ$  và độ lệch lớn nhất là  $10^\circ$  đạt được kết quả như sau:

Xe di chuyển thành công đến mục tiêu đạt 100%.

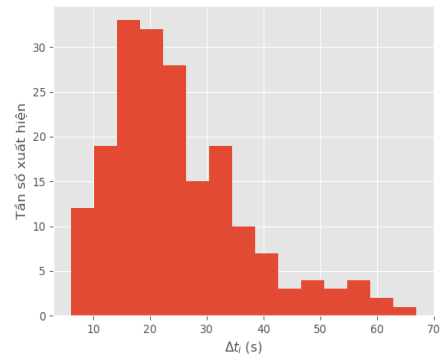
Xe di chuyển thành công đến đúng vị trí chỉ định và có độ lệch khoảng cách dưới 1 m đạt 79% và trên 1 m đạt 96%.

#### TÀI LIỆU THAM KHẢO

Chí, T. N., Văn, N. T. T. (2019). Điều khiển ổn định robot di động đa hướng. *Kỷ yếu Hội nghị Quốc gia lần thứ XXII về Điện tử, Truyền thông và Công nghệ thông tin REV-ECIT 2019* (trang 144-149).

Colledanchise, M., & Ögren, P. (2017). *Behavior trees in robotics and ai: An introduction*. <https://doi.org/10.1201/9780429489105>

Biểu đồ histogram ở Hình 8 thể hiện phân bố tần số xuất hiện các giá trị  $\Delta t_i$ . Qua biểu đồ, có thể thấy  $\Delta t_i$  nằm chủ yếu trong khoảng 5 đến 40 giây. Trong đó các lần thử có  $\Delta t_i$  từ 5 đến 10 giây thuộc những thí nghiệm không bị lỗi do nhiễu định vị và có biên dạng đường đi đơn giản (ví dụ chạy thẳng tới trước như ở thí nghiệm 1). Những lần thử có  $\Delta t_i > 30$  giây là do nhiễu định vị và đáp ứng góc lái không đủ nhanh dẫn đến xe lệch khỏi mục tiêu, làm cây hành vi thực hiện “phục hồi”.



**Hình 8. Phân bố thời gian đến mục tiêu  $\Delta t_i$**

#### 4. KẾT LUẬN VÀ ĐỀ NGHỊ

Nghiên cứu này là một nỗ lực nhằm kết nối giữa nghiên cứu và ứng dụng Robot tự hành. Bài viết cung cấp giải pháp về mô phỏng khảo sát chất lượng của hệ thống tìm đường và bám đường cho xe tự hành lái bằng nguyên lý Ackermann trên ROS và Gazebo. Kết quả khảo sát cho thấy hệ thống đã đưa Robot thành công đến các vị trí chỉ định trong tất cả các thí nghiệm đặt ra với sai số vị trí nhỏ hơn 1,9 m và sai số hướng đầu xe nhỏ hơn  $10^\circ$ .

Nhóm tác giả đề xuất thực hiện thêm nghiên cứu khảo sát chất lượng các hệ thống thường dùng chung với ROS vì hệ thống định vị dựa trên Simultaneous Localization and Mapping và các Cây hành vi còn phức tạp

Coulter, R. C. (1992). *Implementation of the pure pursuit path tracking algorithm*. Carnegie-Mellon UNIV Pittsburgh PA Robotics INST.

Dolgov, D., Thrun, S., Montemerlo, M., & Diebel, J. (2010). Path planning for autonomous vehicles in unknown semi-structured environments. *International Journal of Robotics Research*, 29(5), 485–501. <https://doi.org/10.1177/0278364909359210>

- Dubins, L. E. (1957). On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *American Journal of Mathematics*, 79(3), 497. <https://doi.org/10.2307/2372560>
- Elverum, C. W. (2022). *NTNU Open: Creating the brain of Staale - A navigation system for an autonomous Ackermann-steered litter robot*. <https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/3019934>
- Foote, T. & Purvis, M. (2010). *Standard units of measure and coordinate conventions*. <https://www.ros.org/repos/rep-0103.html>
- Hiệp, Đ. Q., Tiến, N. M., Cường, N. M., Thăng, L. T., & Minh, P. X. (2020). Xây dựng hệ thống nhận thức môi trường cho robot tự hành omni bốn bánh dựa trên thuật toán EKF-SLAM và hệ điều hành ROS. *Tạp chí Nghiên cứu KH&CN quân sự, Số Đặc san Hội thảo Quốc gia FEE, 10 – 2020*, 30-37.
- Hiếu, L. T., Thành, V. C., Nhựt, T. N. M., & Nguyễn, N. K. (2022). Thiết kế mô hình SLAM robot bốn bánh chủ động sử dụng ros. *TNU Journal of Science and Technology*, 227(11), 42–49. <https://doi.org/10.34238/tnu-jst.6113>
- Hường, T. T. (2022). Nghiên cứu điều khiển robot tự hành ứng dụng cho điều hướng thông minh trên cơ sở thuật toán q-learning. *TNU Journal of Science and Technology*, 227(08), 291–300. <https://doi.org/10.34238/tnu-jst.5745>
- Khánh, N. T. (2022). *Beetle (GitHub repository)*. <https://github.com/khanh93vn/beetle>
- Lee, D.-H. (2022). Design of Teleoperation System for Ackermann-steering Mobile Robot considering Driving Stability. *IEMEK Journal of Embedded Systems and Applications*, 17(2), 109–114. <https://doi.org/10.14372/IEMEK.2022.17.2.109>
- Macenski, S., Foote, T., Gerkey, B., Lalancette, C., & Woodall, W. (2022). Robot Operating System 2: Design, architecture, and uses in the wild. *Science Robotics*, 7(66), eabm6074. <https://doi.org/10.1126/scirobotics.abm6074>
- Moore, T., & Stouch, D. (2014). A Generalized Extended Kalman Filter Implementation for the Robot Operating System. *Proceedings of the 13th International Conference on Intelligent Autonomous Systems (IAS-13)*.
- Moore, T. (2016a). *Configuring robot\_localization*. [http://docs.ros.org/en/noetic/api/robot\\_localization/html/configuring\\_robot\\_localization.html](http://docs.ros.org/en/noetic/api/robot_localization/html/configuring_robot_localization.html)
- Moore, T. (2016b). *Integrating GPS Data*. [http://docs.ros.org/en/noetic/api/robot\\_localization/html/integrating\\_gps.html](http://docs.ros.org/en/noetic/api/robot_localization/html/integrating_gps.html)
- Open Robotics. (2022). *ROS 2 Documentation*. <https://docs.ros.org/en/humble>
- Qin, B., Yan, H., Zhang, H., Wang, Y., & Yang, S. X. (2021). Enhanced reduced-order extended state observer for motion control of differential driven mobile robot. *IEEE Transactions on Cybernetics*, 1–12. <https://doi.org/10.1109/tycb.2021.3123563>
- Reeds, J. A., & Shepp, L. A. (1990). Optimal paths for a car that goes both forwards and backwards. *Pacific Journal of Mathematics*, 145(2), 367-393.
- Štefek, A., Pham, V. T., Krivanek, V., & Pham, K. L. (2021). Optimization of fuzzy logic controller used for a differential drive wheeled mobile robot. *Applied Sciences*, 11(13), 6023. <https://doi.org/10.3390/app11136023>
- Tấn, N. & Phúc, N. H. (2021). *Thiết kế và chế tạo xe tự hành* (Luận văn tốt nghiệp). Đại học Cần Thơ.
- Thái, P. M. (2005). Tính năng dẫn hướng của ô tô – máy kéo. Trong N. H. Cần (2005), *Lý thuyết ô tô máy kéo* (trang 230-269). Nhà xuất bản Khoa học và kỹ thuật.