

DESIGN ROS-BASED SLAM MOBILE ROBOT FOUR WHEELS DRIVE

Luu Trong Hieu*, Vo Cong Thanh, Trinh Nguyen Minh Nhut, Nguyen Khac Nguyen

College of Engineering Technology, Can Tho University

ARTICLE INFO		ABSTRACT
Received:	04/6/2022	The topic of mobile robot is still interested by many researchers because of the flexible on moving and searching unknown environment. In recent years, Slam with robot operating system (ROS) is one of the main research topics. However, many researchers simulate by software or use an industrial mobile robot. This paper aims to introduce a design method of a Slam mobile robot with four wheels drive. Overall system includes: i) a low-level-control system and ii) a high-level-control system. The low-level-control system controls and sends robot's velocity to the high-level-control while the high-level-control gathers environment information from Lidar sensor, maps and calculates the mobile robot kinematics. Testing includes mapping an indoor room and mapping a maze. Results show that Slam works well with high accuracy. Besides, controller can control robot moving to any position in the map.
Revised:	24/6/2022	
Published:	24/6/2022	
KEYWORDS		
Slam Robot		
Four wheels drive		
ROS		
Navigation system		
Mapping		

THIẾT KẾ MÔ HÌNH SLAM ROBOT BỐN BÁNH CHỦ ĐỘNG SỬ DỤNG ROS

Luu Trọng Hiếu*, Võ Công Thành, Trịnh Nguyễn Minh Nhựt, Nguyễn Khắc Nguyên

Khoa Công nghệ, Trường Đại học Cần Thơ

THÔNG TIN BÀI BÁO		TÓM TẮT
Ngày nhận bài:	04/6/2022	Lĩnh vực robot di động vẫn là chủ đề được nhiều nhà khoa học quan tâm do khả năng di chuyển, khám phá và tìm hiểu những địa điểm mà con người không tới được. Trong đó, Slam với hệ điều hành robot (robot operating system - ROS) đang là xu hướng phát triển chính. Tuy nhiên, các nghiên cứu đa số tập trung vào mô phỏng thuật toán hoặc dựa vào các mô hình robot hai bánh đã bán trên thị trường. Nghiên cứu này đề xuất thiết kế một mô hình robot bốn bánh chủ động có thể hoạt động như một Slam dựa vào ROS. Toàn bộ hệ thống bao gồm: i) bộ điều khiển cấp thấp và ii) bộ điều khiển cấp cao. Trong đó bộ điều khiển cấp thấp điều khiển và gửi thông tin vận tốc về bộ cấp cao. Bộ điều khiển cấp cao thu thập dữ liệu môi trường từ cảm biến Lidar, xây dựng bản đồ và thiết lập bài toán động học từ dữ liệu nhận về. Kết quả thử nghiệm tại phòng thí nghiệm và một mê cung nhân tạo cho thấy robot hoạt động tốt, khi có thể xây dựng được bản đồ và diện tích vùng quan sát được có diện tích chính xác cao. Bên cạnh đó, robot cũng có thể di chuyển trong bản đồ theo yêu cầu của người điều khiển.
Ngày hoàn thiện:	24/6/2022	
Ngày đăng:	24/6/2022	
TỪ KHÓA		
Slam Robot		
Bộ lái 4 bánh chủ động		
ROS		
Hệ thống định vị		
Xây dựng bản đồ		

DOI: <https://doi.org/10.34238/tnu-jst.6113>

* Corresponding author. Email: luutronghieu@ctu.edu.vn

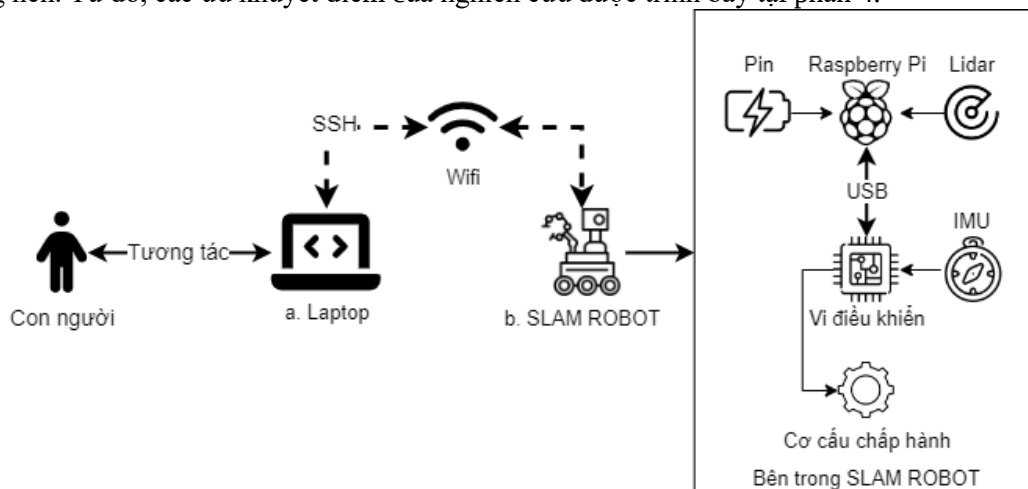
1. Giới thiệu

Ngày nay, nhiều ứng dụng thực tế của Robot di động đã được áp dụng trên toàn thế giới. Robot di động có thể được tìm thấy trong hệ thống logistic để mang và cất hàng hóa, dẫn đường, giám sát và kiểm soát chất lượng. Trong nghiên cứu, bài toán Slam (The Simultaneous Localisation and Mapping) là một lĩnh vực chính tuy nhiên vẫn còn khá mới mẻ của robot học. Các nghiên cứu tiên phong về hệ robot thông minh do Leonard [1] trình bày vào năm 1990. Tiếp theo đó 2 nghiên cứu về SLAM robot được giới thiệu bởi Durrant-Whyte và Bailey [2], [3] trở thành nền móng cơ bản cho các nghiên cứu về SLAM trên toàn thế giới. Hiện nay, xu hướng nghiên cứu robot di động tập trung vào các sử dụng ROS [4] – [7]. Các nghiên cứu này tập trung vào mô phỏng thuật toán hoặc sử dụng các robot di động có sẵn trên thị trường như Pioneer hoặc Turtlebot, điều này thật sự không phù hợp với môi trường Việt Nam khi kinh phí nghiên cứu còn hạn hẹp.

Tại Việt Nam, các nghiên cứu về Robot di động vẫn là chủ đề được tập trung nghiên cứu. Thiết kế bộ điều khiển cho robot di động được nghiên cứu bởi nhóm tác giả [8], [9]. Các nghiên cứu này tập trung thiết kế bộ điều khiển mờ cho robot di động hoặc bộ điều khiển định vị vị trí từ một hệ máy ảnh trên trần nhà. Đặc điểm các nghiên cứu này là môi trường biết trước được, không xuất hiện nhiều chướng ngại vật. Bên cạnh đó, các nghiên cứu về SLAM cũng được tiến hành tại Việt Nam [10], [11]. Các nghiên cứu này tập trung xây dựng các giải thuật phát triển dựa trên mô hình robot di động có sẵn, đa số các mô hình này đều là mô hình xe 2 bánh, với mô hình động học là không đổi. Do đó, các giải pháp điều khiển di chuyển, tính toán vận tốc là tương đối giống nhau.

Nghiên cứu này trình bày một giải pháp thiết kế mô hình Slam robot bằng hệ điều hành ROS. Robot di động được thiết kế bằng phần mềm vẽ chuyên dụng và in 3D để lắp đặt các mạch điều khiển và cơ cấu chấp hành. Bộ điều khiển cấp thấp bao gồm 1 vi điều khiển STM32F4 điều khiển 4 động cơ chủ động thông qua các mạch công suất L298. Bộ điều khiển cao cấp bao gồm một laptop và một máy tính nhúng Raspberry pi 4, đều được cài đặt hệ điều hành ROS. Robot di động có khả năng tự định vị, tự lập bản đồ và tự di chuyển để cập nhật bản đồ. Sau khi quá trình dựng bản đồ kết thúc, người điều khiển có thể điều khiển robot di chuyển đến vị trí mong muốn.

Cấu trúc bài báo bao gồm: phần 1 trình bày các nghiên cứu quốc tế nổi bật cũng như các xu hướng nghiên cứu chính tại Việt Nam. Phần 2 trình bày phương pháp thiết kế phần cứng, bộ điều khiển cấp thấp bao gồm PID và vận tốc cho các bánh chuyển động. Tiếp đó, bộ điều khiển cấp cao trình bày phương pháp định vị và xây dựng bản đồ được hỗ trợ bởi hệ điều hành ROS. Phần 3 trình bày kết quả thực nghiệm tại phòng thí nghiệm và mô hình mô phỏng nhân tạo do nhóm xây dựng nên. Từ đó, các ưu khuyết điểm của nghiên cứu được trình bày tại phần 4.



Hình 1. Sơ đồ tổng quan của hệ thống

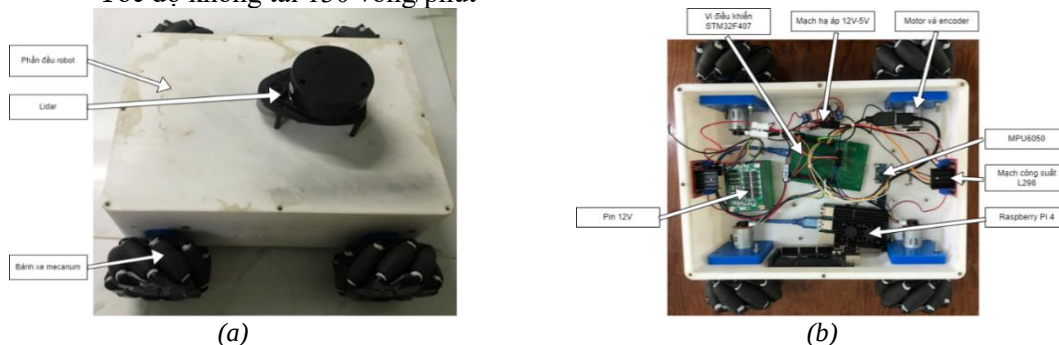
2. Phương pháp thiết kế

Sơ đồ tổng quan của hệ thống được trình bày tại hình 1. Người dùng sử dụng máy tính để điều khiển, quan sát kết quả tạo bản đồ và ra lệnh cho robot di chuyển đến vị trí thông qua máy tính nhúng Raspberry pi 4. Cả hai thiết bị được cài đặt hệ điều hành ROS trên Ubuntu 20.04 và kết nối chung trong một hệ thống wifi, giao tiếp qua chuẩn SSH, trong đó Raspberry được cấu hình IP tĩnh. Máy tính nhúng nhận tín hiệu về từ cảm biến Lidar để xây dựng bản đồ và giao tiếp với vi điều khiển STM32F4 thông qua chuẩn USB. Vi điều khiển nhận tín hiệu từ IMU và gửi về máy tính nhúng, đồng thời điều khiển cơ cấu chấp hành gồm 4 động cơ DC dựa trên bộ điều khiển PID.

2.1. Thiết kế cơ khí

Toàn bộ khung xe được in 3D, có kích thước lần lượt là $280 \times 200 \times 100$ mm, và trục động cơ có $\phi = 36\text{mm}$ (hình 2a). Do mục tiêu của xe chỉ di chuyển trong phòng thí nghiệm nơi có địa hình bằng phẳng với độ trượt bánh thấp, bánh xe được sử dụng là dạng bánh đa hướng mecanum với bán kính $r = 48\text{mm}$. Khung robot được tính toán và thiết kế với phần nắp xe bao gồm một cảm biến Lidar RPLidar A1, phần thân xe bao gồm một máy tính nhúng Raspberry Pi 4, một mạch vi điều khiển MTS32F4, hai mạch công suất L298 và bốn động cơ DC GA-25 có gắn encoder hai kênh A, B (hình 2b) với các thông số kỹ thuật như sau:

- Điện áp hoạt động 12 (V).
- Dòng không tải 0,15 (A).
- Dòng chịu đựng tối đa khi có tải 0,75 (A).
- Tốc độ không tải 130 vòng/phút



Hình 2. Mô hình Robot (a) bên ngoài, (b) bên trong

2.2. Quá trình SLAM của Robot di động

Đối với SLAM robot, quá trình xây dựng bản đồ trong một môi trường không xác định là yếu tố quan trọng nhất. Quá trình SLAM bao gồm nhiều bước: xác định mốc, thu thập dữ liệu khoảng cách trong môi trường, ước tính các giá trị trạng thái, liên kết dữ liệu lại, di chuyển và cập nhật vị trí và trạng thái mới cho robot. Tổng hợp SLAM là một quá trình trong đó mỗi phần đơn của quá trình đó được thực hiện bằng nhiều thuật toán khác nhau. Toàn bộ quá trình SLAM trong nghiên cứu này gồm các bước sau:

- Thiết lập một hệ trục tọa độ cố định đối với môi trường quan sát.
- Thiết lập một hệ trục tọa độ di động nằm trên robot.
- Thiết lập phương trình động học cho Robot (vận tốc điều khiển các bánh và quãng đường di chuyển được)
- Tính toán vị trí của robot và vật thể được quan sát với hệ trục tọa độ cố định dựa trên tín hiệu từ cảm biến Lidar.
- Kết nối dữ liệu và xây dựng bản đồ quan sát.
- Cập nhật vị trí, so sánh thay đổi và nâng cấp bản đồ quan sát.

Quá trình di chuyển của Robot trong bản đồ được thực hiện từ cả 2 bộ điều khiển cấp thấp và cấp cao. Trong đó bộ điều khiển cấp thấp bao gồm: điều khiển PID cho động cơ và vận tốc cho các bánh khi di chuyển và đọc góc xoay từ IMU của robot. Bộ điều khiển cao cấp tính toán sự thay đổi vị trí của robot theo thời gian (Odometry) dựa trên các tín hiệu nhận về từ vi điều khiển và cảm biến lidar thông qua thư viện đóng gói robot_localization từ ROS.

2.2.1. Thiết lập bộ điều khiển PID cho động cơ

Để điều khiển robot chuyển động, bộ điều khiển PID được thiết lập cho 4 động cơ lái bánh xe. Tín hiệu ngõ vào là số xung xác lập (vòng/phút), giá trị ngõ ra là vận tốc động cơ dưới dạng độ rộng xung, tín hiệu hồi tiếp về là số xung đo đặc từ encoder (vòng/phút).

Phương trình PID cho hệ động cơ:

$$RPM = K_p * Err + K_I * integral + K_D * derivative \quad (1)$$

$$Err = set_{RPM} - RPM$$

$$prev_error = error$$

$$derivative = Err - prev_error \quad (2)$$

$$integral = integral + error$$

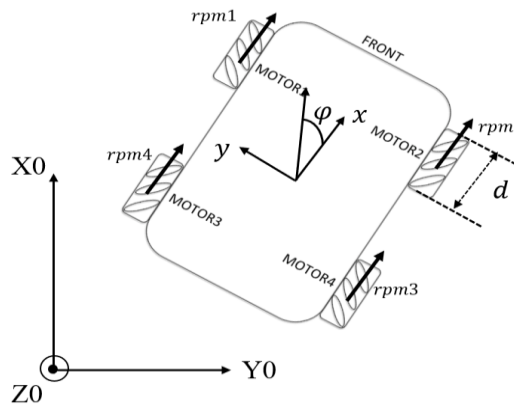
Trong đó:

- RPM: số vòng trên phút.
- Err , $prev_error$: chênh lệch số vòng trên phút của động cơ trong khoảng thời gian t.
- set_{RPM} : số xung xác lập.

Bằng phương pháp thử và sai, hệ số PID lần lượt là $K_p = 0.6$, $K_I = 0.3$, $K_D = 0.5$.

2.2.2. Mô hình động học

Mô hình động học của robot di động sử dụng 4 bánh xe chủ động trong hệ tọa độ Descarte được trình bày tại hình 3.



Hình 3. Mô hình động học của robot di động 4 bánh chủ động

Mô hình động học của SLAM robot trong nghiên cứu này được miêu tả như sau:

- 1- Thiết lập một hệ trục tọa độ cố định đối với môi trường quan sát: vi điều khiển gửi tọa độ $0(X_0, Y_0, Z_0)$ về ROS.
- 2 – Thiết lập hệ tọa độ cho Robot di động (x, y, φ) .
- 3 – ROS gửi vận tốc xuống cho Robot. Phương trình biến đổi vận tốc dài thành vận tốc của 4 bánh xe được trình bày như sau:

Vận tốc tiếp tuyến:

$$v_a = v_z \times \left(\frac{L_x}{2} + \frac{L_y}{2} \right) \quad (3)$$

Số vòng quay trên phút theo phương (x, y, φ) lần lượt:

$$rpm_x = \frac{v_x}{d} \quad (4)$$

$$rpm_y = \frac{v_y}{d} \quad (5)$$

$$rpm_\varphi = \frac{v_\varphi}{d} \quad (6)$$

Số vòng quay của từng motor là:

$$rpm1 = rpm_x - rpm_y - rpm_\varphi \quad (7)$$

$$rpm2 = rpm_x + rpm_y + rpm_\varphi \quad (8)$$

$$rpm3 = rpm_x + rpm_y - rpm_\varphi \quad (9)$$

$$rpm4 = rpm_x - rpm_y + rpm_\varphi \quad (10)$$

4 – Robot gửi thông số vận tốc về ROS một cách liên tục thông qua phương trình sau:

$$v_x = \left(\frac{rpm1 + rpm2 + rpm3 + rpm4}{r} \right) \times \pi d \quad (11)$$

$$v_y = \left(\frac{-rpm1 + rpm2 + rpm3 - rpm4}{r} \right) \times \pi d \quad (12)$$

$$av_\varphi = \left(\frac{-rpm1 + rpm2 - rpm3 + rpm4}{r} \right) \times \pi d \quad (13)$$

$$\varphi = \left(\frac{av_\varphi}{\frac{L_x}{2} + \frac{L_y}{2}} \right) \quad (14)$$

Trong đó :

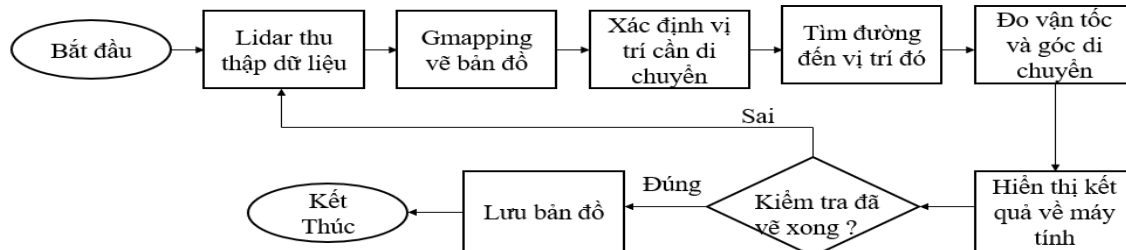
- v_a, v_x, v_y là các giá trị vận tốc nhận được từ ROS (m/s), tối đa 0,5 m/s.
- r, d : bán kính và đường kính bánh xe (mm).
- L_x, L_y : khoảng cách giữa 2 bánh xe theo phương x và phương y (mm).
- rpm_{ix}, rpm_{iy} : tốc độ và hướng quay theo phương (x, y) của động cơ thứ i . Giá trị dương là quay thuận chiều, âm quay ngược chiều (vòng/phút).
- $rpm(i)$: số vòng quay của từng motor (vòng/phút).

2.2.3. Định vị (navigation) và xây dựng bản đồ (mapping)

ROS hỗ trợ đa dạng trong bài toán định vị: định vị cục bộ (Simultaneous localization) và định vị toàn cục (Simultaneous globalization) dựa trên các gói hàm thư viện đóng gói (package). Global_planner là hàm thư viện hỗ trợ giải quyết bài toán toàn cục. Phương pháp này sử dụng thuật toán A* để tìm đường [12]. Dwa_local_planner (Dynamic Window Approach - DWA) [13] là hàm thư viện cung cấp bộ điều khiển robot di chuyển trong mặt phẳng. Bộ điều khiển này kết nối kết quả của bài toán tìm đường đến robot. Khi có bản đồ, bộ tìm đường sẽ tạo quỹ đạo động học để robot đi từ vị trí đầu đến đích. Trên đường đi, bộ tìm đường sẽ tạo một bản đồ lưới nhỏ xung quanh robot và điều khiển robot di chuyển theo các ô lưới này. Công việc của bộ điều khiển là xác định vận tốc dài và vận tốc góc để gửi tới robot.

Sau khi tinh chỉnh các thông số cho hai hàm thư viện toàn cục và cục bộ, quá trình SLAM của robot được bắt đầu. Lúc này robot sẽ vừa di chuyển vừa tính toán các bài toán vận tốc, định vị và xây dựng bản đồ. Quá trình xây dựng bản đồ của robot được thiết kế như Hình 4. Quy trình xây dựng bản đồ tự động được thực hiện như sau: đầu tiên robot sẽ thu thập toàn bộ môi trường chưa xác định thông qua cảm biến Lidar. Các tọa độ này sẽ được sử dụng để vẽ bản đồ hiện tại bằng phương pháp Gmapping [14], từ đây những vị trí còn thiếu sót trên bản đồ sẽ là vị trí Robot cần di chuyển đến để hoàn thiện bản đồ [15]. Robot sẽ tự động di chuyển đến vị trí được xác định,

trong suốt giai đoạn di chuyển, dữ liệu được cập nhật thời gian thực về máy tính, từ đó người quan sát có thể kiểm tra được tình hình Robot. Khi đến vị trí mới, quá trình thu thập dữ liệu mới sẽ được thực hiện. Quá trình này chỉ kết thúc khi Lidar không thể thu thập được dữ liệu mới, hay nói cách khác bản đồ đã được hoàn thiện. Lúc này, quá trình SLAM đã kết thúc, người điều khiển có thể điều khiển Robot đến bất kỳ vị trí nào trong bản đồ theo yêu cầu.



Hình 4. Quy trình tự động xây dựng bản đồ

3. Kết quả đạt được

Để đánh giá khả năng vận hành của phần mềm, Robot được thiết kế để chạy trong 2 thí nghiệm: môi trường trong nhà và mê cung nhân tạo.

3.1. Thực nghiệm trong phòng thí nghiệm

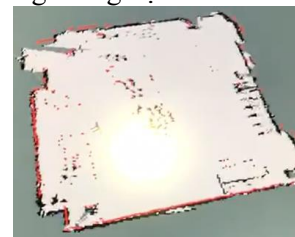
Thí nghiệm được bố trí tại phòng thí nghiệm Hệ thống thông minh, bộ môn Tự động hóa, Khoa Công nghệ, trường Đại học Cần Thơ. Phòng thí nghiệm có diện tích $8 \times 8 \text{ m}^2$, nền lát gạch, địa hình di chuyển phù hợp với Robot đã thiết kế. Khi tiến hành thí nghiệm, phòng trống không có sinh viên, robot di chuyển trong địa hình trống trải. Vì vậy, quá trình Slam của robot diễn ra trong khoảng 2 phút. Hình 5a miêu tả phòng thí nghiệm, hình 5b,c cho thấy quá trình Slam và hoàn thành bản đồ. Bản đồ thu được tại hình 5(c) là một tập hợp các đám mây điểm (point cloud) do cảm biến Lidar quét và tạo ra. Đây là tập hợp vị trí các chướng ngại vật (chấm màu đỏ) và vùng không gian trống trong không gian 2D được biểu thị dưới dạng tọa độ (x, y, φ) so với tọa độ gốc $0(X0, Y0, Z0)$. Đường màu đỏ thể hiện cho đường biên (boundary) của phòng thí nghiệm. Diện tích vùng quan sát được đo lại trên phần mềm, kết quả cho thấy toàn bộ diện tích vùng quan sát là $7,53 \times 7,76 \text{ m}^2$, có sai lệch so với diện tích thực của phòng. Nguyên nhân có thể do các tủ được đặt sát tường, từ đó làm nhỏ đi diện tích sát của phòng thí nghiệm.



(a)



(b)



(c)

Hình 5. Quá trình Slam bản đồ phòng thí nghiệm (a) Hiện trường phòng thí nghiệm (b) bắt đầu quá trình dựng bản đồ (c) hoàn thành bản đồ

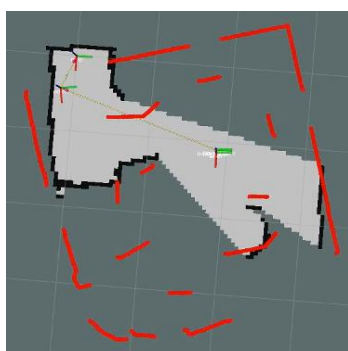
3.2. Thực nghiệm trong mê cung nhân tạo

Mê cung nhân tạo được xây dựng như hình 6a với diện tích $4 \times 4 \text{ m}^2$. Mê cung có nhiều vị trí khuất đòi hỏi robot phải di chuyển để tìm vị trí mới. Robot được bố trí tại vị trí kín ở góc mê cung, một máy ảnh thương mại được sử dụng để ghi lại toàn bộ quá trình Slam của robot. Do đây là mê cung tương đối phức tạp, robot phải di chuyển tới 3 vị trí khác nhau để hoàn thành bản đồ (Hình 6b, c). Diện tích bản đồ sau khi hoàn thành là $3,88 \times 3,96 \text{ m}^2$, kết quả cho thấy độ chính xác cao hơn so với thí nghiệm 1. Sau khi hoàn thành bản đồ, người điều khiển có thể điều khiển vị trí

robot đi chuyển đến trong mê cung (hình 6d), mũi tên hồng biểu diễn cho phần đầu của xe hướng tới sau khi thực hiện quá trình di chuyển. Sau khi xác định được điểm đích trên bản đồ, ROS sẽ tính toán quỹ đạo di chuyển và hiển thị lên màn hình điều khiển bằng đường xanh lá cây (hình 6e).



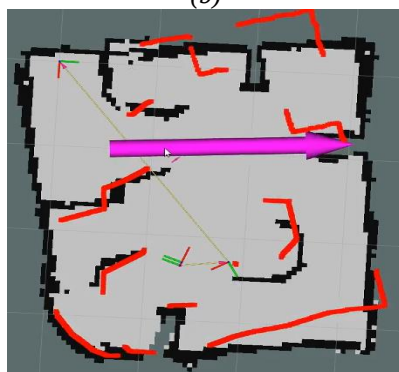
(a)



(b)



(c)



(d)



(e)

Hình 6. Quá trình Slam mê cung nhân tạo (a) Hình mê cung nhân tạo thực tế (b), (c) quá trình Slam và hoàn thành bản đồ, (d) định vị vị trí cho robot di chuyển (e) quỹ đạo di chuyển

4. Kết luận và đề nghị

Nghiên cứu này thiết kế một mô hình SLAM robot 4 bánh chủ động phục vụ cho nghiên cứu trong phòng thí nghiệm. Phần cứng được thiết kế trên phần mềm chuyên dụng trước khi được in 3D và lắp đặt. Phần mềm bao gồm 1 máy tính có cài đặt ROS cho người dùng quan sát trạng thái robot và một vi máy tính Raspberry Pi 4 có cài đặt ROS. Mô hình động học cho Robot được đề xuất để tính toán vận tốc cho 4 bánh xe, gửi dữ liệu về ROS để đo đặc quãng đường đi được của Robot trong suốt quá trình vận hành. Quá trình định vị và xây dựng bản đồ của robot được tiến hành thông qua các gói dữ liệu được hỗ trợ bởi ROS.

Hai thí nghiệm được tiến hành để đánh giá độ hiệu quả của phương pháp. Kết quả cho thấy Robot có thể xây dựng được bản đồ vùng quan sát, có thể đánh dấu được các chướng ngại vật lên bản đồ. Ngoài ra, người điều khiển có thể ra lệnh cho Robot di chuyển đến vị trí mong muốn.

Thời gian tới, nhóm sẽ nghiên cứu tích hợp vào hệ điều khiển 1 máy ảnh thương mại, từ đó robot SLAM sẽ thành hệ kết hợp nhiều cảm biến, nâng cao chất lượng dò tìm và quan sát trong môi trường chưa xác định được.

Phụ lục clip minh họa

<https://www.youtube.com/watch?v=ypD9KEYNQ-0>

TÀI LIỆU THAM KHẢO/ REFERENCES

- [1] J. Leonard, H. Durrant-Whyte, and I. J. Cox, "Dynamic map building for autonomous mobile robot," *IEEE International Workshop on Intelligent Robots and Systems, Towards a New Frontier of Applications*, vol. 1, pp. 89-96, 1990.
- [2] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part I," *IEEE Robotics & Automation Magazine*, vol. 13, no. 2, pp. 99-110, June 2006.
- [3] T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping (SLAM): part II," *IEEE Robotics & Automation Magazine*, vol. 13, no. 3, pp. 108-117, Sept. 2006.
- [4] I. Z. Ibragimov and I. M. Afanasyev, "Comparison of ROS-based visual SLAM methods in homogeneous indoor environment," *14th Workshop on Positioning, Navigation and Communications (WPNC)*, 2017, pp. 1-6.
- [5] Y. Abdelrasoul, A. B. S. H. Saman, and P. Sebastian, "A quantitative study of tuning ROS gmapping parameters and their effect on performing indoor 2D SLAM," *2nd IEEE International Symposium on Robotics and Manufacturing Automation (ROMA)*, 2016, pp. 1-6.
- [6] S. Park and G. Lee, "Mapping and localization of cooperative robots by ROS and SLAM in unknown working area," *56th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, 2017, pp. 858-861.
- [7] M. Rojas-Fernández, D. Mújica-Vargas, M. Matuz-Cruz, and D. López-Borreguero, "Performance comparison of 2D SLAM techniques available in ROS using a differential drive robot," *International Conference on Electronics, Communications and Computers (CONIELECOMP)*, 2018, pp. 50-58.
- [8] T. H. Luu, L. H. Le, and H. H. Nguyen, "Design and simulation of fuzzy control for mobile robot," (in Vietnamese), *Journal of Science and Technology – the University of Da Nang*, vol. 1, no. 86, pp. 48-51, 2015.
- [9] T. A. Tran, H. H. Nguyen, T. H. Luu, and L. H. Le, "A navigation and analysis system using autonomous mobile robot with ceiling camera and onboard micro-spectrometer," (in Vietnamese), *Journal of Science and Technology – the University of Da Nang*, vol. 11, no. 86, pp. 48-51, 2016.
- [10] A. H. Phan, N. D. Bui, D. Q. H. Tran, D. T. Vu, and V. T. T. Nguyen, "Research and development a mapping and navigation simultaneously using ROS system" (in Vietnamese), *The 23rd National Conference on Electronics, Communications and Information Technology*, 2020.
- [11] M. B. Dinh, V. A. Dang, T. C. Nguyen, and X. V. Hoang, "Evaluation and Optimization of Hector SLAM Algorithm for mapping and navigation on Pimouse Robot," (in Vietnamese), *The 24th National Conference on Electronics, Communications and Information Technology*, 2021.
- [12] E. Marder-Eppstein, E. Berger, T. Foote, B. Gerkey and K. Konolige, "The Office Marathon: Robust navigation in an indoor office environment," *IEEE International Conference on Robotics and Automation*, 2010, pp. 300-307.
- [13] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 23-33, March 1997.
- [14] G. Grisetti, C. Stachniss, and W. Burgard, "Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters," *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 34-46, Feb. 2007.
- [15] H. Jiří. "Map-merging for multi-robot system," Bachelor Thesis, Charles University in Prague, Faculty of Mathematics and Physics, 2016.