

# Bài toán SLAM cho hệ thống tự lái ở trường đại học

Lê Đức Anh

Khoa Kỹ thuật Điện & Điện tử, Đại học Bách khoa Thành phố Hồ Chí Minh  
Thành phố Hồ Chí Minh, Việt Nam  
anh.leanhle44@hcmut.edu.vn

**Tóm tắt**—Trong lĩnh vực robot, việc tìm giải pháp cho vấn đề định vị và lập bản đồ đang trở thành xu hướng tất yếu. Simultaneous Localization and Mapping (SLAM) đang nổi lên như một trong những mục tiêu hàng đầu. Các phương tiện tự lái sẽ trở nên phổ biến trong khuôn viên trường đại học, tham gia vào các nhiệm vụ như giao nhận thức ăn, vận chuyển tài liệu. Xây dựng bản đồ 2D trong nhà và ngoài trời, cùng việc định vị phương tiện tự lái trên đó, là một thách thức lớn. Việc này không chỉ đòi hỏi sự hiểu biết về công nghệ mà còn liên quan mật thiết đến kiến thức toán học và các thuật toán phức tạp. Trong bài báo này sẽ không chỉ nghiên cứu lý thuyết của SLAM mà còn giới thiệu cách triển khai và các thuật toán áp dụng để giải quyết vấn đề này. Điều này nhấn mạnh sự cần thiết của việc kết hợp sâu sắc giữa kiến thức lý thuyết và ứng dụng thực tiễn trong việc phát triển SLAM và các ứng dụng của nó.

**Từ khóa**—AGV, Navigation, SLAM

## I. GIỚI THIỆU CHUNG

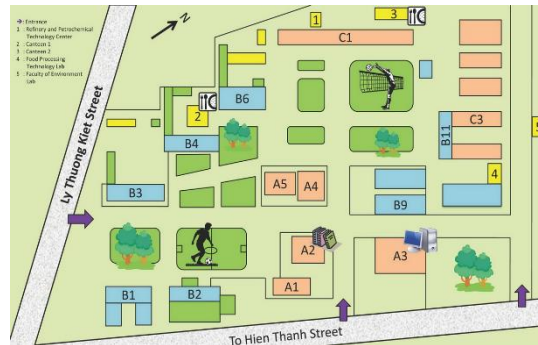
Simultaneous Localization and Mapping (SLAM) là một lĩnh vực quan trọng trong ngành robotics và thị giác máy tính. Lĩnh vực này liên quan đến khả năng của robot hoặc hệ thống tự hành để đồng thời xác định vị trí của chúng trong môi trường không biết và xây dựng bản đồ của môi trường đó.

Ở Việt Nam, các bài toán về SLAM thường thấy trên xe tự hành. Nổi bật nhất phải kể đến chiếc xe tự hành thông minh cấp độ 4 đầu tiên ở Việt Nam đã chính thức được giới thiệu tại hội thảo quốc tế "Giao thông thông minh và Công nghệ tự động hóa tại Việt Nam" do Tập đoàn Phenikaa tổ chức. Việc xây dựng bản đồ 3D cho chiếc xe này không chỉ dùng những phương pháp SLAM thông thường mà còn dùng các phương pháp học sâu.



Hình. 1. Xe tự hành thông minh cấp độ 4 đầu tiên ở Việt Nam

Trong bài báo lần này, bài toán đặt ra ở đây là xây dựng được một bản đồ 2D dành cho khu vực trong nhà trường Đại Học Bách Khoa cơ sở Lý Thường Kiệt và xác định được vị trí vật thể cần được định vị trên toàn bộ khu vực trường.



Hình. 2. Sơ đồ khu vực trường Đại Học Bách Khoa cơ sở Lý Thường Kiệt.

Trong quá trình xây dựng bản đồ 2D, tồn tại những rủi ro đáng chú ý mà nhà nghiên cứu và các nhà phát triển phải đối mặt. Đầu tiên, dữ liệu đầu vào từ Lidar, mặc dù cung cấp thông tin chính xác về môi trường xung quanh, nhưng có thể chứa đựng nhiễu từ môi trường. Các yếu tố như ánh sáng mạnh, bóng cây, hoặc các vật thể động có thể làm mờ dữ liệu thu được, làm giảm tính chính xác của bản đồ. Thứ hai, việc quét bản đồ cần phải xem xét thời gian thích hợp để tránh ảnh hưởng từ hoạt động người qua lại và các yếu tố khác. Đặc biệt, trong môi trường ngoài trời, việc sử dụng GPS để định vị có thể gặp nhiều thách thức. Điều này có thể do sự che phủ của đám mây, cây cối, hoặc các cấu trúc khác, gây ra sự giảm độ chính xác trong việc xác định vị trí. Những vấn đề này đòi hỏi sự kỹ lưỡng trong việc lựa chọn phương pháp và công nghệ, cũng như việc phát triển các thuật toán để xử lý các rủi ro này, nhằm tạo ra các bản đồ 2D chính xác và tin cậy trong nhiều điều kiện môi trường khác nhau.

## II. CÁC THUẬT TOÁN VÀ MÔ HÌNH DÙNG TRONG BÀI TOÁN

### A. Mô hình toán học của SLAM

Trong bài toán hiện tại, Automated Guided Vehicles (AGV) bắt đầu di chuyển từ một vị trí đã biết tọa độ trước, và nó di chuyển ngẫu nhiên trong môi trường không xác định. Sự chuyển động của AGV là không chắc chắn vì nó không thể xác định vị trí tiếp theo, do đó, việc xây dựng bản đồ chính xác trở nên ngày càng khó khăn. Bài toán SLAM là bài toán xây dựng bản đồ môi trường đồng thời xác định vị trí của AGV trong bản đồ đã được xây dựng.

Bài toán SLAM sử dụng các thuật toán xác suất để ước lượng quỹ đạo di chuyển của AGV và xây dựng bản đồ thời gian thực của môi trường xung quanh. Theo xác suất, bài toán SLAM được chia thành hai mô hình là full SLAM và online SLAM.

Giả sử AGV hoạt động trong môi trường không xác định. Gọi  $t$  là thời điểm mà ta đang xét với  $0 \leq t \leq T$ . Khi đó, vị trí của AGV tại thời điểm  $t$  là  $x_t$ , khi AGV chuyển động trên mặt phẳng,  $x_t$  sẽ được mô tả bằng vectơ 3 chiều bao gồm hai giá trị tọa độ của AGV trong 2D và một giá trị quay của AGV. Tập hợp các vị trí của AGV như sau:

$$X_t = \{x_0, x_1, x_2, \dots, x_T\} \quad (1)$$

Với  $x_0$  là vị trí ban đầu của AGV đã được xác định và  $T$  là thời điểm kết thúc trong quá trình tính toán.

Gọi  $u_t$  là giá trị đặc trưng (mối liên hệ) cho chuyển động giữa hai vị trí của AGV hay một chuỗi các phép đo hình học thu được trong quá trình di chuyển. Tập hợp các  $u_t$  như sau:

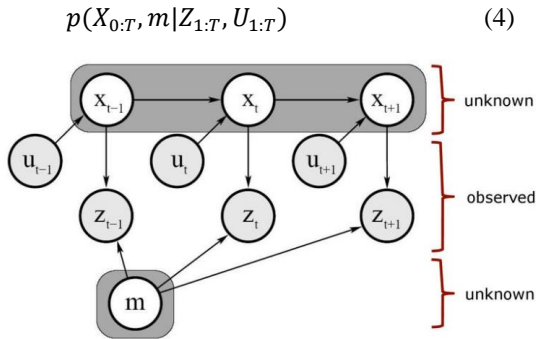
$$U_t = \{u_1, u_2, u_3, \dots, u_T\} \quad (2)$$

Gọi  $z_t$  là giá trị đặc trưng cho thông tin vị trí thông qua các phép đo quan sát từ môi trường  $m$  (định vị). Tập hợp các  $z_t$  như sau:

$$Z_t = \{z_1, z_2, z_3, \dots, z_T\} \quad (3)$$

Với  $m$  là đặc trưng cho bản đồ đúng của môi trường mà AGV di chuyển trong đó.

Mô hình SLAM được chia làm hai loại: mô hình full SLAM và mô hình online SLAM. Đối với mô hình online SLAM chỉ tính toán dữ liệu về vị trí và bản đồ ở thời điểm hiện tại mà không giữ lại cả quá trình di chuyển. Ngược lại, full SLAM tập trung vào việc tính toán toàn bộ quãng đường di chuyển từ lúc bắt đầu đến thời điểm hiện tại của AGV cùng với bản đồ của môi trường. Cụ thể, mô hình full SLAM là mô hình giúp xác định tập hợp tất cả các vị trí  $X_t$  và xây dựng bản đồ của môi trường  $m$  thông qua phép đo hình học  $U_t$  và dữ liệu môi trường  $Z_t$ . Được biểu diễn qua biểu thức:



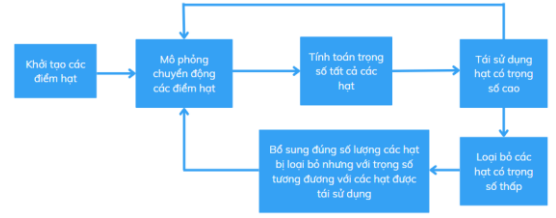
Hình. 3. Mô hình full SLAM.

Do đó, mô hình full SLAM phù hợp với việc xây dựng một bản đồ 2D và xác định được vị trí vật thể cần được định vị.

#### B. Thuật toán Adaptive Monte Carlo Localization (AMCL)

Adaptive Monte Carlo Localization (AMCL) là một thuật toán xác suất sử dụng bộ lọc hạt (particle filter) để ước lượng vị trí 2D của AGV dựa trên dữ liệu cảm biến.

Thuật toán này được phát triển dựa trên thuật toán Monte Carlo Localization (MCL). Thuật toán MCL được chia thành 6 bước như sau:



Hình. 4. Các bước trong thuật toán MCL.

Khi khởi tạo vị trí của AGV là hoàn toàn không chắc chắn khi chưa thực hiện phép quan sát với môi trường. Do đó, thuật toán MCL sẽ được sử dụng để lấy mẫu vị trí từ mô hình chuyển động, đánh giá trọng số bằng mô hình quan sát và tiến hành lấy các mẫu quan trọng thông qua các trọng số [9]. Việc tính toán trọng số của mỗi hạt dựa trên mức độ phù hợp so với bản đồ; sau đó, ước lượng vị trí và định vị AGV. Tuy nhiên, thuật toán MCL xuất hiện vấn đề khi tại một số thời điểm vị trí thay đổi không liên tục dẫn đến việc xác định vị trí của AGV sẽ không chính xác vì khi robot di chuyển một quãng đường lớn hơn vùng lân cận hoặc bị lỗi trong quá trình định vị toàn cục thì robot không thể khôi phục lại vị trí từ tập particle trước đó [2]. Tuy nhiên, thuật toán AMCL có thể giải quyết vấn đề đó bằng cách thêm ngẫu nhiên các hạt tự do ở bước 6 trong thuật toán MCL [5], số lượng các hạt được tính toán dựa trên các trọng số ước tính dài hạn  $\omega_{slow}$  và trọng số ước tính ngắn hạn  $\omega_{fast}$  được biểu diễn dưới biểu thức sau:

$$\begin{cases} \omega_{slow} = \omega_{slow} + \alpha_{slow}(\omega_{avg} - \omega_{slow}), \\ \omega_{fast} = \omega_{fast} + \alpha_{fast}(\omega_{avg} - \omega_{fast}). \end{cases} \quad (5)$$

Trong đó,  $\omega_{avg}$  là trọng số trung bình của tất cả các điểm hạt;  $\alpha_{slow}$  và  $\alpha_{fast}$  là tham số dùng để ước tính tốc độ suy giảm của bộ lọc hàm mũ với trọng số dài hạn và ngắn hạn ( $\alpha_{fast} \gg \alpha_{slow} \geq 0$ ).

Ở bước tái lấy mẫu ở thuật toán AMCL, thuật toán Kullback-Leibler divergence (KLD) được sử dụng, thuật toán này có thể tính toán dễ dàng số lượng hạt cần thiết dựa trên phân phối trọng số của các hạt. Cận trên  $N_{top}$  của số lượng hạt được biểu diễn như sau:

$$N_{top} = \frac{k-1}{2\alpha} \left(1 - \frac{2}{9(k-1)} + \sqrt{\frac{2}{9(k-1)}}\beta\right)^3 \quad (6)$$

Trong đó,  $\alpha$  và  $\beta$  là sai số lớn nhất và biến ngẫu nhiên phân phối chuẩn giữa phân phối thực tế và phân phối xác suất;  $k$  là số lượng trạng thái của tất cả các hạt ( $k \neq 0$ ).

Qua biểu thức (5),  $N_{top}$  của số lượng hạt gần như có quan hệ tuyến tính với  $k$ . Trong quá trình định vị toàn cục ban đầu, các hạt phân tán nhiều hơn và  $k$  lớn hơn nên  $N_{top}$  của số lượng hạt cao hơn. Khi quá trình định vị toàn cục hoàn tất, vấn đề cần giải quyết lúc này là việc theo dõi quỹ đạo. Trong đó, các hạt có xu hướng tập trung quanh một điểm hay hội tụ hơn và  $k$  nhỏ hơn nên  $N_{top}$  của số lượng hạt giảm. Bằng cách này, số lượng hạt được điều chỉnh hiệu quả hơn. Tổng số hạt được giảm đi nên

thời gian hội tụ ngắn hơn; do đó, thuật toán mang lại hiệu quả tính toán tốt hơn.

### C. Thuật toán tối ưu đồ thị Cartographer

Thuật toán tối ưu đồ thị Cartographer được sử dụng cho bài toán SLAM để giải quyết các vấn đề về vòng lặp phản hồi có thể có các điểm lặp sai khi có nhiễu, các điểm tương tự. Thuật toán sử dụng một bộ dự đoán độ trễ để lọc các điểm lặp bị sai như đã đề cập. Bài toán SLAM sau khi được tối ưu hóa bởi thuật toán Cartographer sẽ được triển khai trên nền tảng robot di động dựa trên ROS và được sử dụng trong ứng dụng tạo bản đồ trong nhà hoặc ngoài trời. Việc sử dụng thuật toán Cartographer trong ứng dụng vẽ bản đồ, có thể thu được bản đồ 2D có độ chính xác tương đối cao ở trong thời gian thực và áp dụng để xây dựng bản đồ trên phạm vi rộng lớn.

Bước đầu tiên của thuật toán là xây dựng bản đồ con. Dữ liệu được quét được chuyển đổi sang hệ tọa độ con bằng phép biến đổi mặt phẳng:

$$T_{\xi} h_k = \begin{bmatrix} \cos \xi_{\theta} & -\sin \xi_{\theta} \\ \sin \xi_{\theta} & \cos \xi_{\theta} \end{bmatrix} h_k + \begin{bmatrix} \xi_x \\ \xi_y \end{bmatrix} \quad (7)$$

Trong đó:  $\xi = (\xi_x, \xi_y, \xi_{\theta})$  là vị trí và hướng của robot trong môi trường 2D.  $\xi_x$  và  $\xi_y$  là độ dịch theo hai hướng x và y,  $\xi_{\theta}$  là góc quay của robot trong mặt phẳng hai chiều x và y,  $H = \{h_k\}_{k=1, \dots, k}, h_k \in \mathbf{R}^2$  là dữ liệu được quét từ lidar,  $T_{\xi}$  là ma trận biến đổi vị trí và hướng trong hệ tọa độ được quét từ lidar với hệ tọa độ con.

Khi dữ liệu mới được quét nó sẽ chèn vào bản đồ con. Khi đó, giá trị xác suất của lưới x được cập nhật dựa vào phương trình sau:

$$\begin{cases} odds(p) = \frac{p}{1-p} \\ M_{new} = clamp(odds^{-1}(M_{old}(x)) \cdot odds(P_{hit})) \end{cases} \quad (8)$$

Trong đó,  $M_{old}$  là giá trị trạng thái trước khi được cập nhật,  $M_{new}$  là giá trị trạng thái sau khi được cập nhật và hàm  $clamp$  là hàm giới hạn phạm vi giá trị xác suất trọng lưới bản đồ.

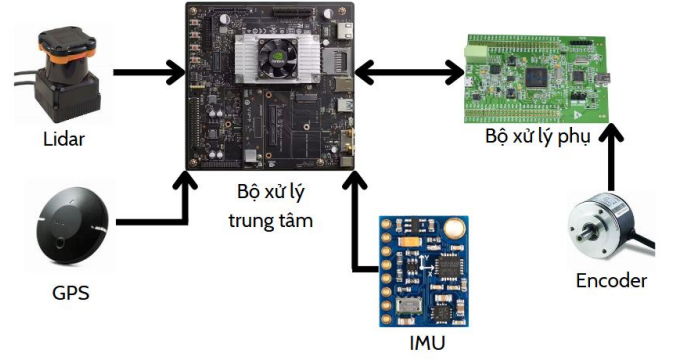
Trước khi chèn dữ liệu mới vào bản đồ con, thuật toán Cartographer sử dụng thư viện Ceres-Solver [4] để thực hiện phép bình phương cực tiểu phi tuyến (phương trình (9)) trên dữ liệu quét và bản đồ con để thu được vị trí và hướng  $\xi$  nhằm tối đa hóa bản đồ đám mây điểm hiện tại sang hệ tọa độ bản đồ con. Phép bình phương cực tiểu phi tuyến được biểu diễn bởi phương trình sau:

$$\underset{\xi}{argmin} \cdot \sum_{k=1}^K (1 - M_{smooth}(T_{\xi} h_k))^2 \quad (9)$$

Trong đó,  $M_{smooth}$  là hàm nội suy bậc 2,  $\mathbf{R}^2 \rightarrow \mathbf{R}$ .

Nhờ thuật toán này mà bản đồ thu được sẽ có độ chính xác cao hơn.

## III. CÁC THÀNH PHẦN TRONG HỆ THỐNG



Hình. 5. Các thành phần trong hệ thống.

### A. Bộ xử lý trung tâm

Bộ xử lý trung tâm sẽ nhận tín hiệu từ các cảm biến thông qua các giao thức như UART, I2C. Bộ xử lý trung tâm ở đây thường dùng là máy tính nhúng. Trong máy tính nhúng sẽ được tích hợp ROS - một hệ thống phần mềm mã nguồn mở được thiết kế đặc biệt cho lập trình robot và các ứng dụng điều khiển tự động, từ đó hệ thống có thể dễ dàng thu thập, xử lý và quản lý dữ liệu từ các cảm biến. ROS cung cấp một cơ chế linh hoạt để tạo ra các node (đơn vị xử lý) độc lập cho việc thu thập dữ liệu từ các cảm biến cụ thể. Các node này có thể giao tiếp với nhau thông qua các topic (chủ đề) trong ROS, cho phép truyền tải dữ liệu giữa chúng một cách dễ dàng và linh hoạt.

### B. Bộ xử lý phụ

Bộ xử lý phụ có chức năng điều khiển driver động cơ cho hệ thống xe. Ngoài ra, bộ xử lý phụ còn đọc encoder và đưa lên cho bộ xử lý chính thông qua giao thức UART nhằm hướng lượng hướng và vị trí cùng với IMU. Bộ xử lý phụ thường là vi điều khiển (ví dụ như Arduino, STM,...).

### C. LiDAR

LiDAR, viết tắt của Light Detection and Ranging, là một phương pháp tiên tiến được sử dụng để đo khoảng cách đến các đối tượng bằng cách gửi tia laser và đo lường sự phản xạ từ chúng thông qua cảm biến. Phương pháp này thường được áp dụng để tạo ra các bản đồ chi tiết với độ phân giải cao của môi trường xung quanh. LiDAR được ứng dụng rộng rãi trong nhiều lĩnh vực như trắc địa, địa tin học, khảo cổ học và các công nghệ kỹ thuật khác.

Ở đây, dữ liệu LiDAR thu thập để đăng lên chủ đề trong ROS là dạng PointCloud, tức là một tập hợp các điểm 3D trong không gian, mỗi điểm được xác định bằng các tọa độ (x, y, z) và thông tin khác như màu sắc, độ phản xạ và khoảng cách. Khi thu thập dữ liệu LiDAR, các điểm này được ghi lại trong khung tọa độ của cảm biến.

### D. IMU

IMU, viết tắt của Inertial Measurement Unit, là một công cụ đo lường quán tính được tích hợp trong một loạt các thiết bị từ máy bay, AGV cho đến các chiếc điện thoại thông minh. Khả năng đo lường chính xác và đáng tin cậy của IMU là kết quả của sự kết hợp giữa hai thành phần chính: gia tốc kế và con quay hồi



chuyển. Gia tốc kế đo lường các thay đổi về gia tốc, đại diện cho sự thay đổi về vận tốc hoặc hướng của một vật thể. Trong khi đó, con quay hồi chuyển ghi nhận các biến đổi về góc quay của thiết bị. Bằng cách kết hợp hai thành phần này, IMU có thể cung cấp thông tin chi tiết và toàn diện về chuyển động của các thiết bị di động trong không gian ba chiều.

Đối với IMU, thông tin để đưa lên topic của ROS bao gồm các giá trị đo tốc độ góc (angular velocity), gia tốc (acceleration), và có thể bao gồm cả dữ liệu về hướng quay (orientation) của đối tượng. Vì các thông số đọc IMU rất nhiều nên phải đưa qua một vài bộ lọc. Bộ lọc thường được dùng nhất là bộ lọc Kalman.

#### E. GPS

GPS (Global Positioning System) là một hệ thống phức tạp dựa trên vị trí của các vệ tinh nhân tạo, được phát triển, vận hành và quản lý bởi Bộ Quốc phòng Hoa Kỳ. Tính đến một thời điểm nhất định, GPS có thể xác định vị trí của một điểm trên mặt đất bằng cách tính toán khoảng cách từ điểm đó đến ít nhất bốn vệ tinh trong hệ thống. Hệ thống GPS không chỉ bao gồm các vệ tinh trong không gian, mà còn có các trạm điều khiển tại mặt đất. Những trạm này chịu trách nhiệm kiểm soát quỹ đạo của các vệ tinh và cung cấp dữ liệu để tính toán vị trí chính xác cho người dùng. Bằng cách sử dụng các trạm cố định như điểm chuẩn, GPS có khả năng cung cấp thông tin vị trí với độ chính xác cao trong hầu hết điều kiện địa lý. Thông tin thu thập từ GPS thường bao gồm các thông số như vĩ độ, kinh độ, độ cao.

Tuy nhiên, trong bài toán định vị không thể dùng trực tiếp vĩ độ, kinh độ mà thường sẽ quy về một hệ quy chiếu x, y khác (thường là trục x là hướng Bắc, trục y là hướng Đông). do đó, ta cần một hàm chuyển đổi để có thể sử dụng được bộ thông số đó với giá trị vĩ độ và kinh độ tham chiếu tương ứng là  $ref\_lat$  và  $ref\_lon$ .

Đầu tiên ta cần biến đổi các giá trị  $lat$  và  $lon$  dựa vào công thức mà nhà sản xuất GPS đưa ra (cách tính của  $lat$  tương tự với  $lon$ ):

Bước 1:  $A = \frac{lat}{100}$ , với A có dạng uint16

Bước 2:  $B = lat - A \times 100$ , với B có dạng double

Bước 3: Cập nhật giá trị  $lat$

$$lat = A + \frac{B}{60}$$

Nếu giá trị  $lat = ref\_lat$  và  $lon = ref\_lon$  thì:

Bước 1: Chuyển đổi bốn giá trị trên về dạng radian, ví dụ

$$lat_{rad} = lat \times \frac{\pi}{180^\circ}$$

Bước 2: Tính giá trị c với biểu thức:

$$c = \cos^{-1}(\cos(ref_{lat}) \times \sin(lat) + \cos(ref_{lat}) * \cos(lat) * \cos(lon_{rad} - ref_{lon_{rad}}))$$

Bước 3: Tính giá trị k.

Ta cho một giá trị  $\varepsilon > 0$

Nếu  $c \leq \varepsilon$  thì  $k = 1$ , ngược lại thì  $k = \frac{c}{\sin(c)}$

Trong quá trình chuyển đổi, biến k được sử dụng để tính toán khoảng cách và hướng từ điểm hiện tại đến điểm tham chiếu. Nó được sử dụng để điều chỉnh việc tính toán, đảm bảo rằng các giá trị được tính toán sẽ chính xác và đáng tin cậy, đặc biệt là trong trường hợp cận giá trị 0, khi có thể gây ra vấn đề với phép chia cho 0.

Bước 4: Tính giá trị x, y là giá trị được đưa vào vi điều khiển

$$x = k \times (ref_{\cos(lat)} \times \sin(lat) - ref_{\sin(lat)} \times \cos(lat) \times \cos(lon_{rad} - ref_{lon_{rad}})) \times 6371000$$

$$y = k \times \cos(lat) \times \sin(lon_{rad} - ref_{lon_{rad}}) \times 6371000$$

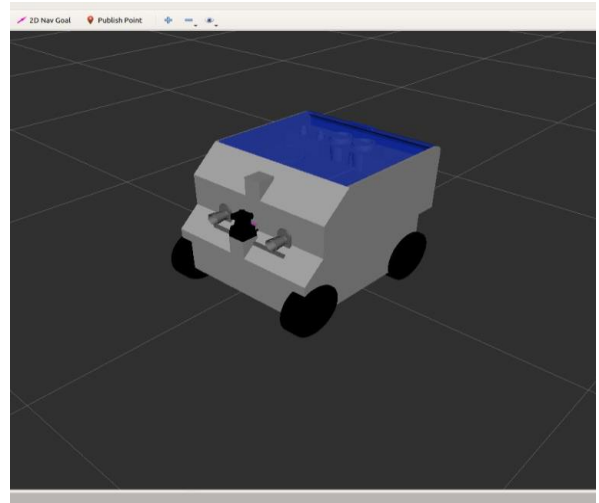
Trong đó: bán kính trái đất là 6371 (km) = 6371000 (m)

#### F. Encoder

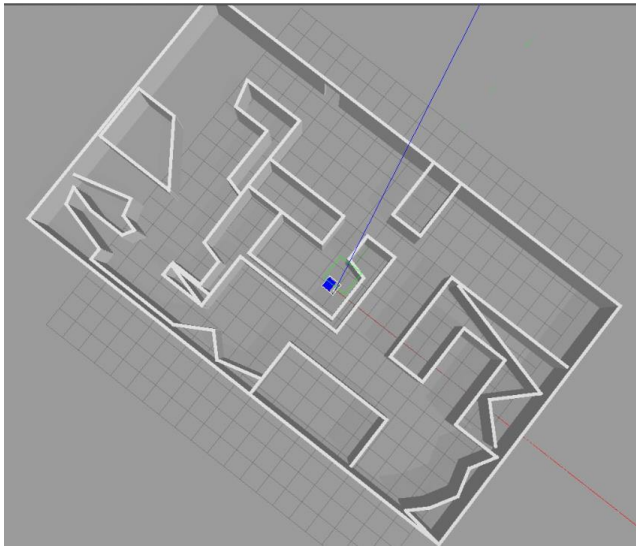
Encoder được sử dụng để thu thập thông tin về tốc độ của AGV hoặc các phương tiện tự động khác trong môi trường. Đối với thông tin encoder được cung cấp tới bộ xử lý trung tâm thông qua bộ xử lý phụ với giao thức UART. Thông số encoder cùng với dữ liệu từ IMU sẽ cung cấp thêm thông tin cho việc dự đoán hướng và vị trí của AGV ở môi trường trong nhà.

### IV. KẾT QUẢ

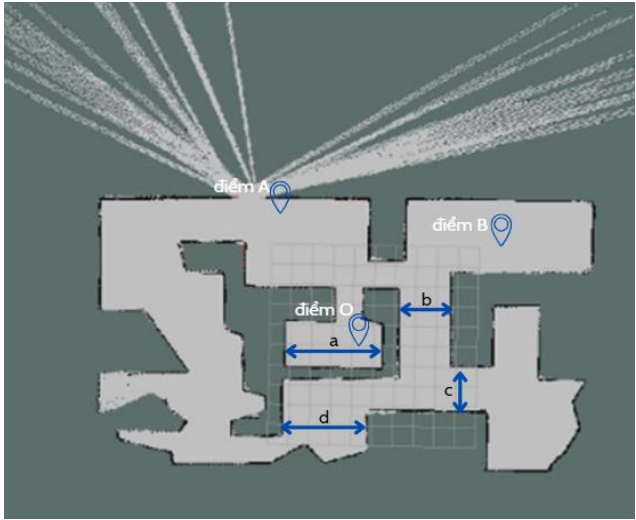
Kết quả mô phỏng trong nhà:



Hình. 6. Mô hình hệ thống tự hành trong mô phỏng



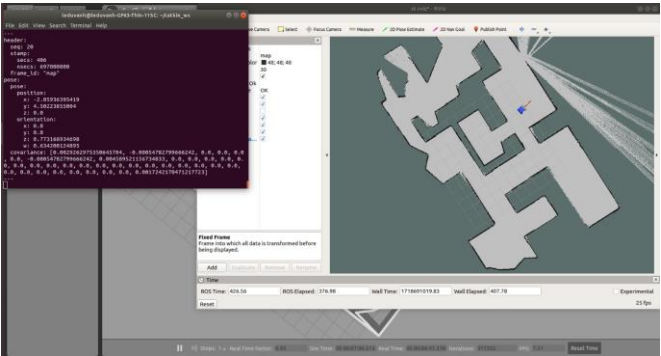
Hình. 7. Bản đồ trong mô phỏng



Hình. 8. Bản đồ offline trong nhà sau khi quét xong bản đồ

BẢNG I. SỐ LIỆU SAI SỐ CỦA BẢN ĐỒ THU ĐƯỢC (MÔ PHỎNG)

Đoạn đường	Kích thước thực tế trên bản đồ mô phỏng(m)	Kích thước trên bản đồ (m)	Sai số (%)
a	4.75	4.78	0.63
b	2.98	3.02	1.32
c	2.04	2.01	-1.49
d	4.25	4.12	-0.71

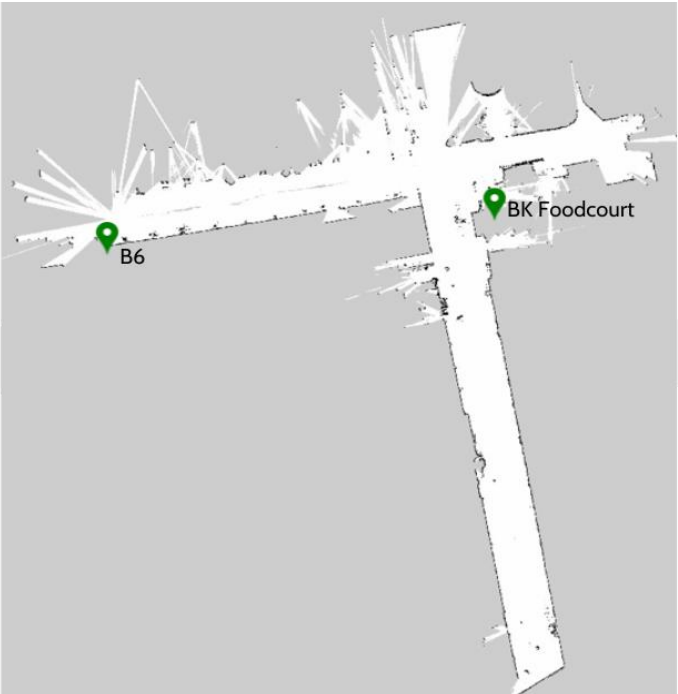


Hình. 9. Định vị bản đồ trên mô phỏng

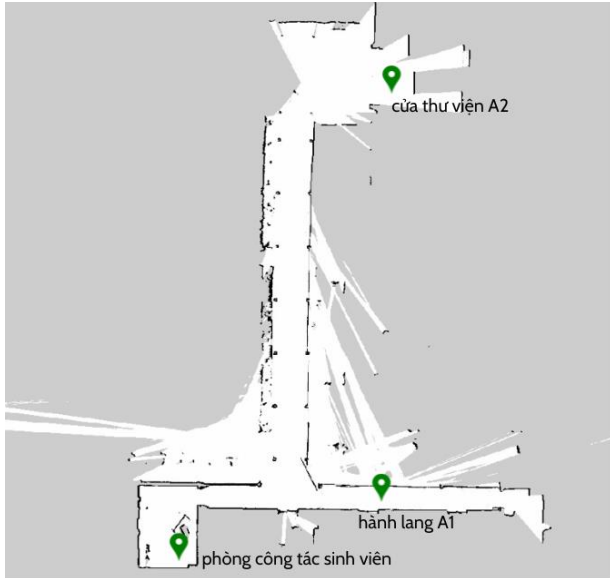
BẢNG II. TỌA ĐỘ ĐỊNH VỊ TRÊN BẢN ĐỒ TRONG NHÀ (MÔ PHỎNG)

Điểm	Tọa độ x	Tọa độ y
O	0.002	-0.001
A	-4.365	6.595
B	6.806	5.649

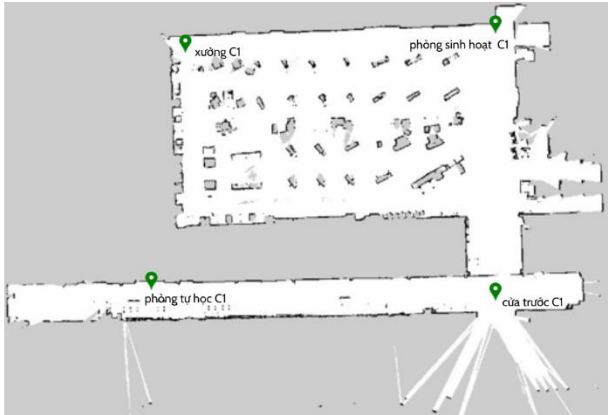
Kết quả thực tế trong nhà:  
Khu vực BK Food court:



Khu vực A2 – A1:



Khu vực xưởng C1:



BẢNG III. TỌA ĐỘ ĐỊNH VỊ TRÊN BẢN ĐỒ TRONG NHÀ (THỰC TẾ)

Điểm	Tọa độ x	Tọa độ y	Sai số thực tế (%)
Cửa thư viện A2	4.012	-0.001	0.91
Phòng công tác sinh viên	-6.235	35.034	3.06
Cửa trước tòa C1	0.002	-0.012	0.014
Phòng tự học C1	0.017	-32.238	2.52
Xưởng C1	33.1289	-33.674	3.78

Kết quả thực tế ngoài trời:

BẢNG IV. TỌA ĐỘ CỦA MỘT SỐ VỊ TRÍ ĐẶC THÙ TRONG BẢN ĐỒ (GNSS)

Tên vị trí	Tọa độ Latitude	Tọa độ Longitude
Xưởng C1	10.7746	106.6595
Tòa A2	10.7729	106.6601
Tòa B6	10.77376	106.65923

## V. KẾT LUẬN

Trong bài báo này, chúng tôi đã trình bày các kết quả thu được từ việc tạo bản đồ và định vị trong nhà cũng như ngoài trời. Kết quả thứ nhất là về định vị trong nhà và tạo bản đồ. Kết quả đạt được độ chính xác cao, với sai số so với thực tế nhỏ hơn 5%. Độ chính xác này phụ thuộc vào độ rộng của bản đồ offline đã quét, đảm bảo rằng bản đồ được tạo ra có độ chính xác tương ứng. Thứ hai là định vị ngoài trời. Tọa độ GPS thu được có độ chính xác thấp, chỉ lấy được 5 chữ số sau dấu thập phân, dẫn đến sai số từ 2 đến 8 mét so với thực tế.

Mặc dù đạt được những kết quả khả quan, nghiên cứu vẫn còn một số hạn chế. Sai số GPS ngoài trời rất lớn, một phần do nhiễu và một phần do không sử dụng các phương pháp cải thiện như RTK (Real-Time Kinematic).

Để cải thiện độ chính xác và hiệu quả của hệ thống định vị, chúng tôi đề xuất một số hướng nghiên cứu tương lai như sau. Thứ nhất là sử dụng các phương pháp cải thiện kết quả Rover: Áp dụng RTK hoặc phương pháp moving base để giảm sai số GPS và nâng cao độ chính xác của tọa độ thu được. Thứ hai là tích hợp thêm thị giác máy trong việc tạo bản đồ. Việc sử dụng thêm các cảm biến như camera màu và camera đo độ sâu có thể cải thiện thông tin và độ chính xác của bản đồ.

Nghiên cứu này đã cung cấp những đóng góp quan trọng trong việc tạo bản đồ và định vị cho các hệ thống tự động trong nhà và ngoài trời. Mặc dù vẫn còn những hạn chế, các đề xuất cải tiến trong tương lai hứa hẹn sẽ nâng cao hiệu quả và độ chính xác của hệ thống, mở ra nhiều ứng dụng tiềm năng trong lĩnh vực robot tự động và dẫn đường.

## LỜI CẢM ƠN

Nghiên cứu được tài trợ bởi Trường Đại học Bách Khoa – ĐHQG-HCM trong khuôn khổ đề tài mã số SVCQ-2023-ĐĐT-107. Tôi xin cảm ơn Trường Đại học Bách Khoa, ĐHQG-HCM đã hỗ trợ thời gian và phương tiện vật chất cho nghiên cứu này.

## TÀI LIỆU THAM KHẢO

- [1] Agarwal S, Mierle K. (2012) Ceres solver. <http://ceres-solver.org>.
- [2] C.-Y. Li, I.-H. Li, Y.-H. Chien, W.-Y. Wang, and C.-C. Hsu, "Improved Monte Carlo localization with robust orientation estimation based on cloud computing," in 2016 IEEE Congress on Evolutionary Computation (CEC), pp. 4522–4527, Vancouver, Canada, July 2016.
- [3] Gili, J.; Corominas, J.; Rius, J. Using global positioning system techniques in landslide monitoring. Eng. Geol. 2000, 55, 167–192.
- [4] Hao J. (2019) SLAM and navigation robot designs based on the Cartographer algorithm. Shandong university.
- [5] L. X. Ding and W. J. Tao, "Design and implementation of localization navigation of indoor mobile robot in unknown environment," Ordnance Automation, vol. 3, 2018.
- [6] Người Việt phát triển xe tự hành cấp độ 4 đầu tiên. (2021) VnExpress. <https://vnexpress.net/nguoi-viet-phat-trien-xe-tu-hanh-cap-do-4-dau-tien-4264713.html>.
- [7] Stopher P, FitzGerald C, Xu M: Assessing the accuracy of the Sydney Household Travel Survey with GPS. Transportation. 2007, 34: 723-741.

