

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**  
**ĐẠI HỌC QUỐC GIA HÀ NỘI**

.....oOo.....



**BÁO CÁO CUỐI KỲ MÔN HỌC LẬP**  
**TRÌNH ROBOT VỚI ROS**

**Đề tài : Tìm hiểu và sử dụng thuật toán Karto Slam**  
**cho robot và xây dựng bản đồ kiểm thử**

Môn học : Lập trình robot với ROS

Mã học phần : 2425II\_RBE3017

Giảng viên bộ môn: Thầy Lê Xuân Lực và thầy Dương Văn Tân

Sinh viên thực hiện : Trương Ngọc Anh - 22027528

Hoàng Văn Cường - 22027549

Bảng phân công nhiệm vụ

	Họ và tên (Full name)	Mã SV (ID)	Đóng góp (Contribution)
Thành viên 1	Trương Ngọc Anh	22027528	Thiết kế và xây dựng map, đánh giá bản đồ về độ khó và kết quả, chạy navigation, làm báo cáo.
Thành viên 2	Hoàng Văn Cường	22027549	Chạy Karto_slam, quét bản đồ, quay video, làm báo cáo.
Địa chỉ Repo trên Github	<a href="https://github.com/truogah831/robot_ros_ck.git">https://github.com/truogah831/robot_ros_ck.git</a>		
Video chạy slam để quét bản đồ	<a href="https://drive.google.com/file/d/1stpKc19Ux9MBt8clcMuUcuAn7psIgsJM/view?usp=drive_link">https://drive.google.com/file/d/1stpKc19Ux9MBt8clcMuUcuAn7psIgsJM/view?usp=drive_link</a> Các bản đồ còn lại: <a href="https://drive.google.com/drive/folders/15tIEXMsxOBhW3xI5I_Dj2i9ViGIyXYXH?usp=drive_link">https://drive.google.com/drive/folders/15tIEXMsxOBhW3xI5I_Dj2i9ViGIyXYXH?usp=drive_link</a>		

# TÓM TẮT

Lập bản đồ là một trong những bài toán cơ bản và quan trọng nhất đối với hệ thống robot di động. Hiện nay, hai hướng tiếp cận phổ biến để giải quyết bài toán này là sử dụng hệ thống cảm biến LiDAR hoặc/và hệ thống cảm biến hình ảnh kết hợp với các thuật toán xử lý dữ liệu. Trong đó, phương pháp sử dụng LiDAR cùng thuật toán Karto SLAM đã thể hiện khả năng xây dựng bản đồ với độ chính xác cao, đặc biệt trong môi trường trong nhà, nhờ ưu thế phát hiện vòng lặp và tối ưu hóa toàn cục quỹ đạo di chuyển.

Tuy nhiên, hiệu suất hoạt động của Karto SLAM chịu ảnh hưởng đáng kể bởi việc lựa chọn và điều chỉnh các tham số như ngưỡng matching, bán kính tìm kiếm vòng lặp, và trọng số cập nhật bản đồ. Để nghiên cứu sâu hơn về tác động của các tham số này, dự án tiến hành triển khai và đánh giá Karto SLAM trên một nền tảng robot di động sử dụng cảm biến LiDAR 2D, vận hành trên hệ điều hành ROS Noetic.

Hiệu năng của hệ thống được đánh giá dựa trên chất lượng bản đồ tạo ra, thông qua các chỉ số về độ phủ và sai số quỹ đạo. Các môi trường thử nghiệm đa dạng như phòng trống, cấu trúc đối xứng và không gian phức tạp được thiết kế nhằm kiểm tra khả năng lập bản đồ của thuật toán trong nhiều điều kiện khác nhau. Bản đồ thu được sẽ được so sánh với bản đồ thiết kế lý thuyết để đánh giá độ chính xác và hiệu quả thực thi của Karto SLAM. Kết quả nghiên cứu giúp làm rõ ảnh hưởng của các tham số cấu hình đến hiệu năng thuật toán, đồng thời cung cấp cơ sở cho việc tinh chỉnh hệ thống theo từng loại robot và môi trường ứng dụng cụ thể.

# LỜI CẢM ƠN

Lời đầu tiên, em xin được gửi lời cảm ơn chân thành và sâu sắc nhất đến TS. Lê Xuân Lực và KS. Dương Văn Tân, những người thầy, người hướng dẫn đã tận tình hỗ trợ, định hướng và đóng góp những ý kiến quý báu trong suốt quá trình nhóm em thực hiện đề tài. Sự nhiệt tình, kiến thức chuyên môn sâu rộng cùng những lời khuyên thiết thực của các thầy đã giúp nhóm em vượt qua nhiều khó khăn, hoàn thiện đề tài một cách tốt nhất.

# Mục Lục

TÓM TẮT .....	3
LỜI CẢM ƠN .....	4
Mục Lục .....	5
I. Giới thiệu .....	6
A. Đặt vấn đề .....	6
B. Mục tiêu dự án.....	6
C. Phạm vi dự án.....	7
D. Cấu trúc báo cáo.....	7
II. Hệ thống robot sử dụng.....	7
A.  Phần mô hình robot.....	7
B.  Karto_slam trong ROS.....	8
Sơ lược về Karto_slam.....	8
Nguyên lý hoạt động của thuật toán .....	9
III. Trình bày các bản đồ kiểm thử và đánh giá kết quả.....	10
A.Điều kiện kiểm thử.....	10
B.Bản đồ kiểm thử .....	10
Bản đồ 1. ....	11
Bản đồ 2. ....	12
Bản đồ 3. ....	14
Bản đồ 4. ....	17
Bản đồ 5. ....	18
Bản đồ 6. ....	20
IV. Navigation cho robot.....	22

A. Kiến trúc hệ thống.....	22
V. Thảo luận và đánh giá.....	25
Trả lời câu hỏi .....	27

## I. Giới thiệu

### A. Đặt vấn đề

SLAM (Simultaneous Localization and Mapping) là một nền tảng công nghệ quan trọng đối với các hệ thống robot tự hành. SLAM cho phép robot vừa xây dựng bản đồ môi trường xung quanh, vừa đồng thời xác định vị trí của chính nó trong bản đồ đó mà không cần dựa vào hệ thống định vị bên ngoài. Đây là yếu tố cốt lõi để robot có thể di chuyển tự chủ trong các môi trường chưa biết.

Tuy nhiên, bài toán SLAM đối mặt với nhiều thách thức kỹ thuật lớn. Một trong những vấn đề nổi bật là sai số tích lũy trong quá trình đo lường odometry, gây lệch lạc quỹ đạo theo thời gian. Bên cạnh đó, sự phức tạp của môi trường thực tế như cấu trúc môi trường đối xứng, nhiều vật cản động hoặc điều kiện ánh sáng thay đổi cũng gây khó khăn cho việc duy trì độ chính xác của bản đồ. Ngoài ra, hạn chế về độ chính xác và tốc độ cập nhật của các cảm biến như LiDAR, camera hay IMU càng làm tăng thêm độ khó trong việc đạt được định vị và lập bản đồ đồng thời một cách ổn định và tin cậy.

Trong bối cảnh đó, việc lựa chọn thuật toán SLAM phù hợp và tối ưu hóa các tham số vận hành là yếu tố then chốt để nâng cao hiệu quả hoạt động của hệ thống robot di động.

### B. Mục tiêu dự án

Dự án này tập trung vào triển khai và đánh giá khả năng ứng dụng của thuật toán Karto SLAM trên nền tảng hệ điều hành robot ROS Noetic. Các mục tiêu cụ thể bao gồm:

- i) Triển khai thuật toán Karto SLAM trên ROS: Thiết lập hệ thống hoàn chỉnh bao gồm cảm biến LiDAR và chạy node slam\_karto để xây dựng bản đồ thời gian thực.
- ii) Tìm hiểu nguyên lý hoạt động của Karto SLAM: Phân tích chi tiết cách Karto SLAM thực hiện xây dựng đồ thị, phát hiện vòng lặp và tối ưu hóa toàn cục.

- iii) Thiết kế và xây dựng môi trường thử nghiệm: Tạo các môi trường thực tế trong nhà với độ khó tăng dần (phòng trống, môi trường đối xứng, không gian phức tạp) để kiểm tra khả năng lập bản đồ của robot.
- iv) Đánh giá hệ thống: Phân tích độ chính xác bản đồ thu được (về diện tích bao phủ, sai số quỹ đạo), kiểm tra hiệu quả phát hiện vòng lặp.

### *C. Phạm vi dự án*

Dự án giới hạn trong các điều kiện thử nghiệm sau:

- Môi trường: Các bài thử nghiệm được thực hiện trong môi trường trong nhà với mặt đất bằng phẳng và môi trường tĩnh, với mặt phẳng hai chiều (2D).
- Cảm biến: Hệ thống sử dụng cảm biến chính là LiDAR 2D để thu thập dữ liệu khoảng cách từ môi trường xung quanh.
- Nền tảng phần mềm: Toàn bộ hệ thống được triển khai trên hệ điều hành Ubuntu, sử dụng ROS Noetic làm nền tảng phát triển và vận hành các node SLAM.

### *D. Cấu trúc báo cáo*

Báo cáo này được tổ chức theo các chương mục chính như sau:

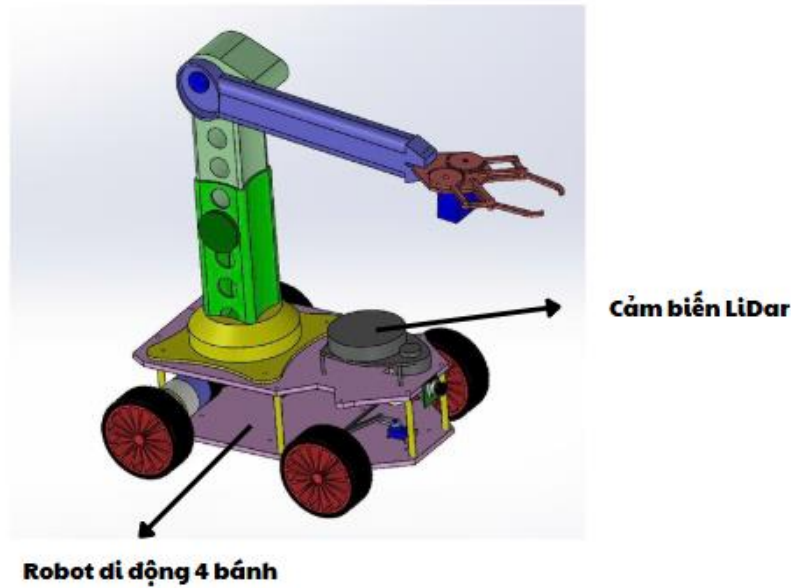
- i) Phần I: Trình bày tổng quan về bài toán SLAM, các thách thức, mục tiêu, phạm vi nghiên cứu và định hướng tổng thể của dự án.
- ii) Phần II: Trình bày về hệ thống robot sử dụng cho dự án.
- iii) Phần III: Trình bày các bản đồ sử dụng để thử nghiệm cùng với đánh giá về bản đồ và phân tích kết quả thử nghiệm, bao gồm các bản đồ thu được, đánh giá chỉ số sai số quỹ đạo, khả năng phát hiện vòng lặp.
- iv) Phần IV: Áp dụng các bản đồ trên để chạy Navigation cho robot
- v) Phần V: Trình bày thảo luận và đánh giá.

## **II. Hệ thống robot sử dụng**

### *A. Phần mô hình robot*

Quá trình kiểm thử và nghiên cứu được thực hiện trên một mô hình robot di động tự thiết kế, sử dụng phần mềm thiết kế cơ khí SolidWorks. Mô hình robot sau khi hoàn thiện được xuất ra định dạng URDF để tích hợp vào môi trường mô phỏng Gazebo.

Trong mô hình, robot được trang bị cảm biến LiDAR 2D, mô phỏng khả năng quét môi trường xung quanh. Các node ROS chịu trách nhiệm thu thập dữ liệu cảm biến, xử lý odometry, và thực thi thuật toán Karto SLAM để xây dựng bản đồ thời gian thực. Hệ thống điều khiển robot bao gồm các node vận tốc (cmd\_vel) trong mô phỏng.



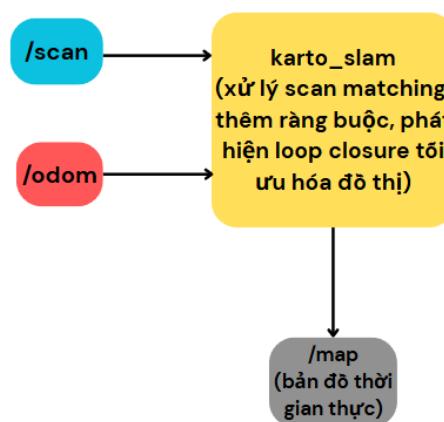
Hình 1: Mô hình robot 4 bánh

### B. Karto\_slam trong ROS

#### Sơ lược về Karto\_slam

Karto SLAM là một thuật toán SLAM 2D thuộc nhóm Graph-based SLAM, được thiết kế nhằm tối ưu hóa quá trình xây dựng bản đồ và định vị robot trong môi trường trong nhà. Thuật toán này nổi bật với khả năng phát hiện vòng lặp hiệu quả.

Trong hệ sinh thái ROS, gói slam\_karto cung cấp một node ROS cài đặt Karto SLAM, cho phép robot sử dụng dữ liệu LiDAR 2D cùng odometry để xây dựng bản đồ thời gian thực. Khác với một số thuật toán như Hector SLAM không yêu cầu odometry, Karto SLAM tận dụng cả thông tin odometry và phép matching giữa các quét LiDAR để cải thiện độ chính xác trong quá trình xây dựng bản đồ.



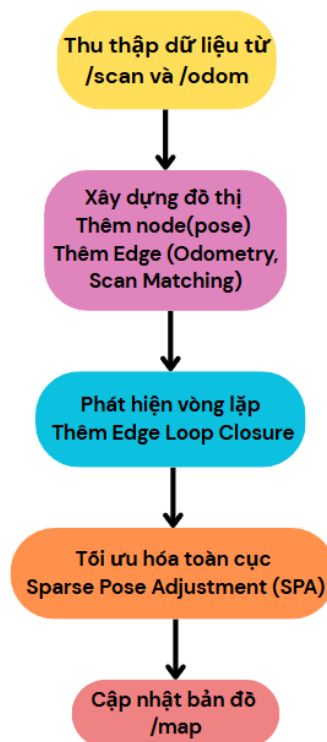
Hình 2: Các node của Karto\_slam



### Nguyên lý hoạt động của thuật toán

Karto SLAM hoạt động dựa trên nguyên lý xây dựng và tối ưu hóa một đồ thị ràng buộc các vị trí của robot. Quá trình vận hành bao gồm các bước chính sau:

- Xây dựng đồ thị: Mỗi vị trí và hướng của robot tại một thời điểm được đại diện bởi một node trong đồ thị. Các cạnh giữa các node thể hiện mối quan hệ ràng buộc, bao gồm:
  - Ràng buộc từ dữ liệu odometry.
  - Ràng buộc từ phép scan matching giữa các quét LiDAR kế tiếp.
- Phát hiện vòng lặp: Khi robot di chuyển và phát hiện mình đang quay lại một vị trí đã đi qua trước đó, một edge loop closure sẽ được thêm vào đồ thị để liên kết các vị trí, giúp giảm thiểu sai số tích lũy.
- Tối ưu hóa toàn cục: Sau khi cập nhật đồ thị, toàn bộ hệ thống sẽ được tối ưu hóa sử dụng kỹ thuật Sparse Pose Adjustment (SPA), nhằm tìm tập hợp poses tốt nhất thỏa mãn tất cả các ràng buộc đã ghi nhận.



Hình 3: Sơ đồ hoạt động thuật toán

Karto SLAM có ưu điểm nổi bật về độ chính xác cao nhờ kết hợp hiệu quả giữa dữ liệu odometry, scan matching và tối ưu hóa toàn cục. Thuật toán đặc biệt phát huy thế mạnh trong các môi trường có cấu trúc lặp lại nhờ khả năng phát hiện và xử lý vòng lặp tốt. Bên cạnh đó, việc triển khai Karto SLAM trên nền tảng ROS cũng rất thuận tiện nhờ sự hỗ trợ sẵn có của gói slam\_karto. Tuy nhiên, Karto SLAM cũng tồn tại một số hạn chế như tiêu tốn tài nguyên

tính toán lớn khi số lượng node và edge tăng, yêu cầu tinh chỉnh tham số phù hợp để đạt hiệu quả tối ưu, và có nguy cơ sai số tích lũy nếu quá trình phát hiện vòng lặp không chính xác.

III. Trình bày các bản đồ kiểm thử và đánh giá kết quả.

A.Điều kiện kiểm thử

Thử nghiệm được tiến hành trong môi trường trong nhà kín với kích thước được mô tả theo bản vẽ kỹ thuật, vật cản ở đây chính là các bức tường xung quanh phòng cùng với các vật cản ngẫu nhiên.

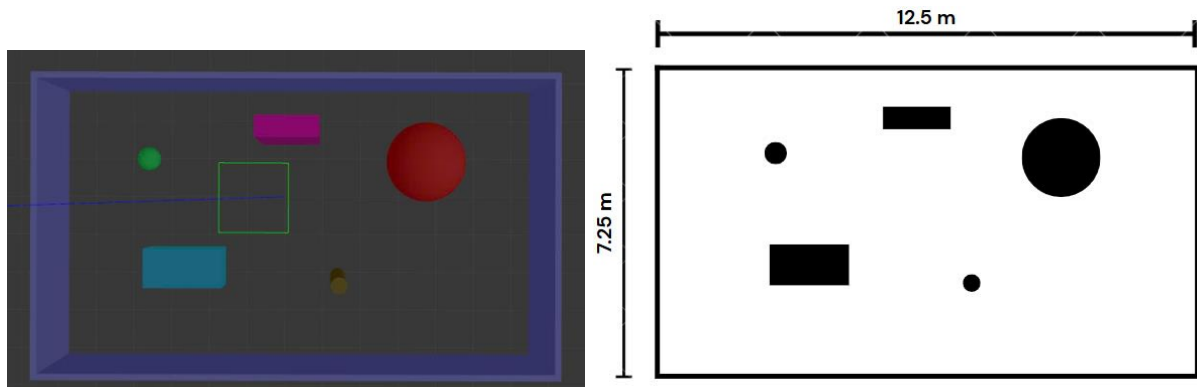
Bảng I. Các tham số cấu hình của hệ thống Karto\_slam

Tham số	Chức năng	Giá trị
map_update_interval	Khoảng thời gian giữa các lần cập nhật bản đồ, nhanh quá thì đòi hỏi lượng tài nguyên tiêu thụ lớn, chậm quá thì không cập nhật được các thay đổi của môi trường.	5.0 giây
minimum_travel_distance	Khoảng cách tối thiểu robot phải di chuyển để thêm node mới vào đồ thị. Tham số quá nhỏ thì sẽ dễ bị ghi nhận cả các node khi robot chỉ di chuyển rất nhỏ hoặc rung lắc, nhưng hữu ích ở những mô trường nhỏ nhiều chi tiết.	0.5 mét
minimum_travel_heading	Góc quay tối thiểu robot phải đạt để thêm node mới. Điều này đặc biệt quan trọng đối với các chuyển động quay tại chỗ robot sẽ không ghi nhận pose mới nếu chỉ lắc nhẹ, giảm độ nhiễu trong quá trình xây dựng đồ thị.	0.5 rad (~28.6 độ)
range_threshold	Giới hạn khoảng cách dữ liệu quét LiDAR được chấp nhận, loại bỏ nhiễu xa.	30.0 mét
resolution	Độ chi tiết của bản đồ; ô nhỏ hơn cho bản đồ sắc nét hơn nhưng tốn tài nguyên hơn.	0.05 mét (5cm/ô lưới)

B.Bản đồ kiểm thử

Tiến hành thiết kế các bản đồ cho việc kiểm thử, xây dựng 6 bản đồ với độ khó tăng dần do sở hữu các đặc điểm gây khó khăn cho slam trong việc quét và xây dựng lại bản đồ.

Bản đồ 1.



Hình 4.1: Bản vẽ kỹ thuật bản đồ kiểm thử 1

Bản đồ 1 thiết kế có kích thước  $12.5\text{m} \times 7.25\text{m}$ , dạng phòng kín với nhiều vật cản bố trí tự do bên trong. Các vật thể có kích thước, hình dạng đa dạng (hình tròn, hình chữ nhật lớn nhỏ khác nhau), phân bố không đối xứng trong không gian.

Các đặc điểm có lợi cho SLAM:

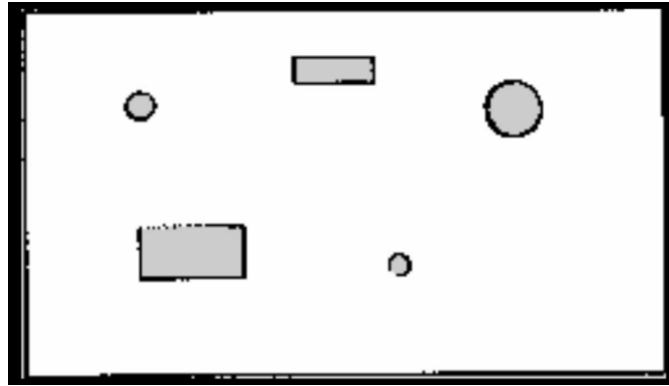
- Không gian tương đối rộng cho phép robot di chuyển tự do, quét được nhiều hướng khác nhau, thuận lợi cho việc thu thập dữ liệu LiDAR liên tục và ổn định.
- Các vật cản có kích thước lớn nhỏ đan xen giúp robot dễ dàng thực hiện scan matching vì luôn có điểm đặc trưng trong tầm quét.
- Không đối xứng hoàn toàn hạn chế khả năng gây nhầm lẫn vị trí trong đồ thị thuận lợi cho thuật toán phát hiện vòng lặp.

Đặc điểm gây khó khăn cho SLAM:

- Vật thể tròn: Các bề mặt cong có thể gây khó khăn cho thuật toán scan matching vốn dựa nhiều vào các đặc trưng góc cạnh.
- Các vật cản nhỏ (hình tròn nhỏ): Các vật thể nhỏ nếu cách xa robot sẽ khó quét chính xác bằng LiDAR tầm thấp, làm giảm độ tin cậy matching hoặc thậm chí bị bỏ qua nếu nằm ngoài ngưỡng `range_threshold`.

Tổng quan: Với một hệ thống SLAM ổn định như Karto SLAM có scan matching và loop closure tốt, bản đồ này ở mức trung bình dễ được sử dụng để kiểm tra khả năng lập bản đồ và xác định vật thể của slam.

Kết quả bản đồ lập được:



*Hình 4.2: Bản đồ 1 lập được nhờ slam*

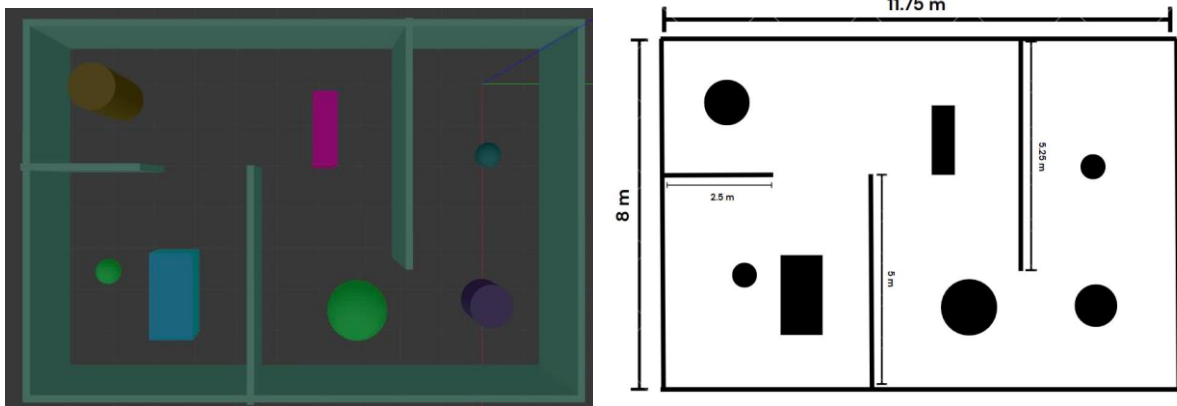
Bản đồ SLAM được xây dựng từ quá trình quét bằng Karto SLAM đã tái hiện tương đối tốt cấu trúc của môi trường thực tế. Các vật thể chính như hình tròn lớn, hình chữ nhật, và các vật thể nhỏ hơn đều đã được nhận dạng và thể hiện đúng vị trí tương đối so với thực tế.

Các bức tường biên ngoài được vẽ thẳng, kín và rõ ràng, cho thấy thuật toán SLAM đã thực hiện tốt việc ghép nối các quét laser trong toàn bộ quá trình di chuyển, các vật thể được nhận diện tốt và không có hiện tượng đứt gãy nghiêm trọng tại các vị trí trống lớn.

Tuy nhiên chi tiết vật thể nhỏ bị mờ nhạt cho thấy độ phân giải scan tại những khu vực này chưa thực sự cao hoặc robot chưa tiếp cận đủ gần.

Nhìn chung thì bản đồ SLAM thu được chính xác ở mức độ tương đối cao và rõ ràng.

*Bản đồ 2.*



*Hình 5.1: Bản vẽ kỹ thuật bản đồ kiểm thử 2*

Bản đồ thiết kế lần này có kích thước  $11.75\text{m} \times 8\text{m}$ , cũng là dạng phòng kín, nhưng phức tạp hơn nhờ sự xuất hiện thêm nhiều vách ngăn bên trong. Các vật thể (hình tròn, hình chữ nhật lớn nhỏ) được phân bố tự do ở nhiều khu vực khác nhau.

Đặc điểm thuận lợi cho SLAM:

- Nhiều vách ngăn phân chia khu vực bán kín, giúp robot luôn có điểm đặc trưng để quét, tăng cường khả năng scan matching chính xác.
- Các vật thể lớn nhỏ được phân bố khắp các khu vực, hạn chế tình trạng robot di chuyển lâu trong vùng trống mà không có thông tin mới.
- Môi trường không đối xứng, giúp thuật toán nhận diện vị trí dễ hơn, giảm nguy cơ nhầm lẫn khi thực hiện loop closure.

Đặc điểm gây khó khăn cho SLAM:

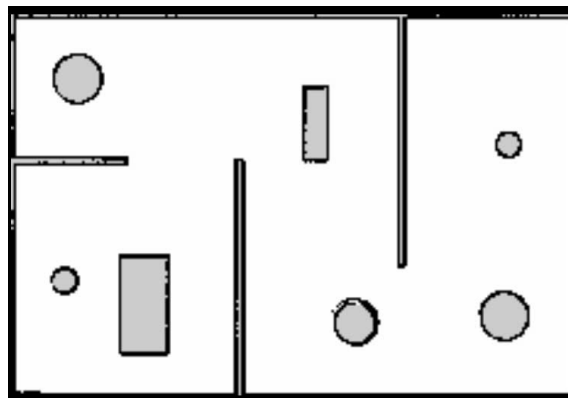
- Hành lang hẹp: Một số đoạn hành lang khá hẹp ( $\sim 1.1\text{m}$ ,  $\sim 1.2\text{m}$ ), dễ dẫn tới:
  - Robot bị hạn chế góc quét laser.
  - Khó khăn trong việc điều khiển khi cua.
  - Scan matching khó khăn do ít điểm dữ liệu trong phạm vi quét.
  - Khó phát hiện chính xác vòng lặp nếu chỉ dựa trên quét trong hành lang.
- Các vật thể hình tròn không tạo ra góc cạnh rõ ràng, có thể gây nhiễu scan matching, nhất là khi chỉ dựa vào một phần mặt cong trong quét.

Tổng quan: Đây là một bản đồ khó hơn đáng kể so với bản đồ trước. Khi thử thách trong các bài toán:

- Scan Matching trong hành lang hẹp.
- Loop Closure Detection trong môi trường bán kín.

Đòi hỏi SLAM phải có khả năng xử lý matching tinh tế và phát hiện vòng lặp thông minh mà Karto SLAM làm tương đối tốt nhờ SPA.

Kết quả bản đồ lập được:



Hình 5.2: Bản đồ 2 lập được nhờ slam

Bản đồ được tạo ra từ quá trình quét SLAM đã mô phỏng khá chính xác cấu trúc môi trường so với thiết kế ban đầu. Các bức tường, hành lang, và các vật cản lớn nhỏ đều đã được nhận diện và thể hiện khá rõ ràng.

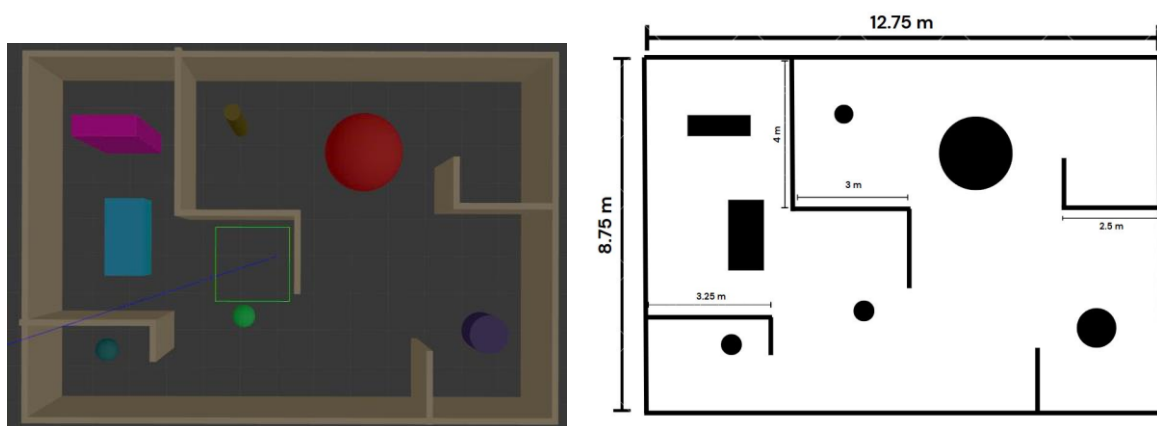
Cấu trúc tổng thể đạt độ chính xác cao: Các tường chính, hành lang phân chia không gian được xây dựng gần như trùng khớp với bản thiết kế, cho thấy quá trình scan matching và tối ưu hóa graph vận hành hiệu quả.

Các vật thể tròn và chữ nhật (bao gồm các vật cản lớn và vừa) được thể hiện rõ ràng, đúng vị trí tương đối, chứng tỏ robot duy trì tốt chất lượng quét trong suốt hành trình.

Không có hiện tượng méo mó bản đồ, không thấy lỗi scan đứt đoạn hay lệch hình nghiêm trọng, cho thấy dữ liệu odometry và LiDAR được sử dụng ổn định.

Tại các vị trí gây khó khăn cho slam như khúc cua hay các hành lang cũng được xử lý chính xác không có hiện tượng méo mó bản đồ, không thấy lỗi scan đứt đoạn hay lệch hình nghiêm trọng, cho thấy dữ liệu odometry và LiDAR được sử dụng ổn định.

*Bản đồ 3.*



*Hình 6.1: Bản vẽ kỹ thuật bản đồ kiểm thử 3*

Bản đồ có kích thước  $12.75\text{m} \times 8.75\text{m}$ , thiết kế với nhiều vách ngăn và phân vùng không gian phức tạp hơn so với hai bản đồ trước. Bản đồ này là cải tiến về độ phức tạp hơn so với bản đồ 2 do có nhiều hơn các khúc cua cùng với các hành lang phức tạp hơn.

Đặc điểm thuận lợi cho SLAM:

- Nhiều vách ngăn tạo vùng đặc trưng giúp robot dễ dàng thực hiện scan matching chính xác nhờ vào việc thu thập được các đặc trưng môi trường khác biệt.
- Các vật thể lớn nhỏ được bố trí rộng khắp, giảm thiểu khoảng trống lớn, giúp robot liên tục có điểm tham chiếu khi di chuyển.
- Cấu trúc bản đồ đa dạng, nhiều vật cản và vách ngăn giúp robot dễ dàng phân biệt các khu vực khác nhau, thuận lợi cho việc phát hiện vòng lặp.

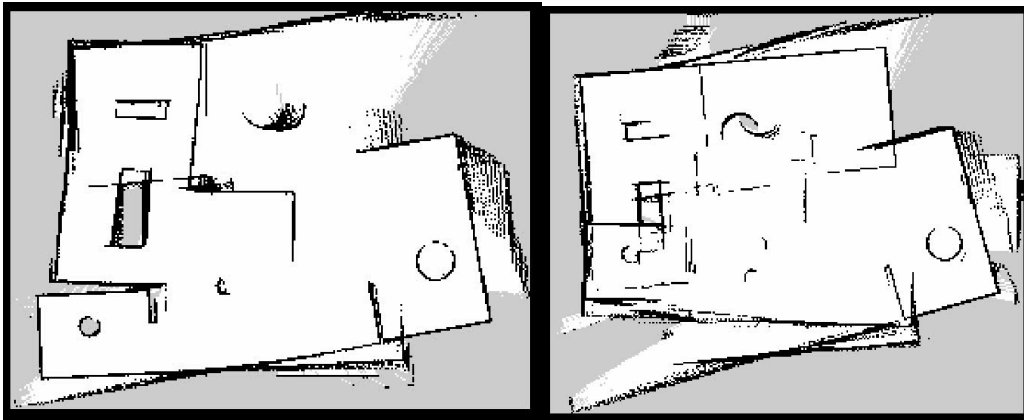
Đặc điểm gây khó khăn cho SLAM:

- Các đoạn hành lang có chiều rộng nhỏ (~2m hoặc thậm chí thấp hơn), robot có thể bị hạn chế góc quét laser, làm giảm số lượng điểm scan đáng tin cậy, từ đó làm tăng nguy cơ sai lệch scan matching.
- Một số vách ngăn che khuất tầm quét LiDAR, gây ra các vùng blind-spot, buộc robot phải định hướng lại nhiều lần để quét đầy đủ, có thể làm tăng độ phức tạp trong việc tối ưu hóa đồ thị.

Tổng quan: Đây là bản đồ khó cho slam do các thách thức về:

- Scan matching liên tục (do các vùng hẹp và khuất tầm nhìn).
- Phát hiện loop closure chính xác (do sự giống nhau giữa các hành lang nhỏ).

Kết quả bản đồ lập được:



*Hình 6.2: Bản đồ lập được sau hai lần đầu quét*

Qua quan sát bản đồ SLAM thu được, có thể thấy bản đồ có các hiện tượng lỗi đặc trưng như:

- Các vùng tường và hành lang bị méo hình, xoay lệch.
- Xuất hiện vết chồng lấn giữa các lần quét.
- Các vật thể hình tròn và vách ngăn không thẳng hàng với tường ngoài, mất tính chính xác hình học ban đầu.
- Bản đồ bị sai lệch.

Phân tích nguyên nhân lỗi theo đặc điểm bản đồ:

Với các phần hành lang hẹp nhỏ khi robot đi vào làm cho góc quét lidar bị hạn chế dẫn đến số lượng điểm đặc trưng giảm làm cho matching yếu dẫn đến tích lũy sai số.

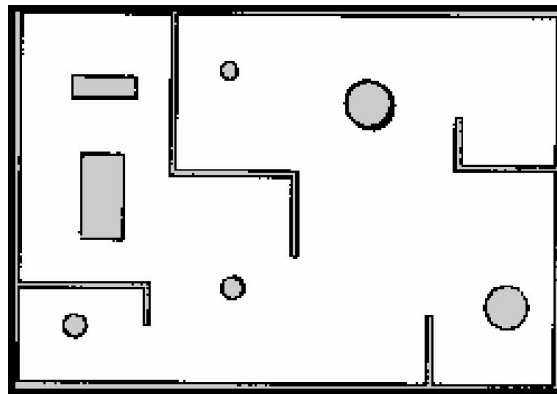
Các điểm mù làm cho hướng quét của lidar bị khuất dẫn đến robot phải dựa hoàn toàn vào odometry dẫn đến sai lệch vị trí theo thời gian.

Cùng với các cấu trúc có hình dạng tương tự làm cho robot bị nhầm lẫn khi phát hiện vòng lặp gây lỗi tối ưu toàn cục.

Bản đồ bị lỗi do kết hợp cả yếu tố địa hình phức tạp (hành lang hẹp, nhiều vách ngăn che khuất) và chưa tối ưu tham số (`minimum_travel_distance` quá cao, `map_update_interval` lớn, `loop closure` yếu). Cải thiện các tham số slam như sau:

- Giảm `minimum_travel_distance` xuống 0.2m để phù hợp với các khúc cua hay địa hình hẹp nhỏ nhiều chi tiết.
- Giảm `map_update_interval` xuống 2giây để cập nhật map nhanh hơn.
- Giảm `resolution` bản đồ xuống 0.03m để chi tiết hóa vật thể nhỏ.
- Tuning thêm các tham số `loop closure` để tăng độ nhạy khi phát hiện vòng lặp trong môi trường phức tạp.
- Kết hợp với điều khiển robot với tốc độ chậm và ổn định hơn.

Kết quả sau khi thay đổi tham số:



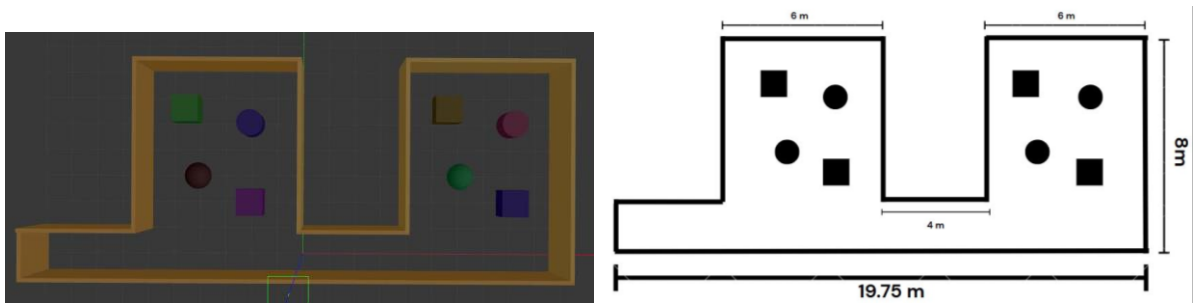
*Hình 6.3: Bản đồ lập được sau lần quét thứ 3 đầu quét*

Bản đồ thu được đã có chất lượng cải thiện đáng kể, tái hiện tương đối chính xác cấu trúc môi trường thực tế, bao gồm hệ thống vách ngăn, hành lang, và các vật thể lớn nhỏ.

Không xuất hiện hiện tượng méo lệch toàn cục: Bản đồ không có dấu hiệu nghiêng lệch hoặc bị vặn xoắn mạnh, thể hiện rằng `loop closure` đã phát hiện và xử lý vòng lặp thành công trong quá trình tối ưu hóa.



Bản đồ 4.



Hình 7.1: Bản vẽ kỹ thuật bản đồ kiểm thử 4

Bản đồ lần này có kích thước khá lớn,  $19.75\text{m} \times 8\text{m}$ , với thiết kế dạng hành lang chữ U, phân thành hai khu vực đối xứng, mỗi bên chứa nhiều vật cản rải rác.

Đặc điểm thuận lợi cho SLAM:

- Hành lang rộng: Đoạn hành lang ngang và các khu vực rộng (từ 4m trở lên) giúp robot di chuyển dễ dàng và thu thập dữ liệu quét tốt. Góc quét LiDAR được đảm bảo bao phủ rộng rãi.
- Các vật cản hình tròn và hình vuông được bố trí đều đặn trong mỗi khu vực, tạo ra nhiều điểm đặc trưng, giúp thuật toán scan matching hoạt động ổn định.

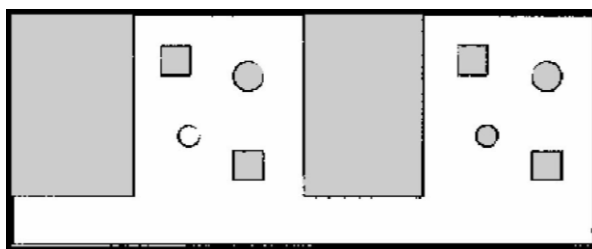
Đặc điểm gây khó khăn cho SLAM:

- Khu vực hành lang dài và trống: Phần hành lang dài có thể gây hiện tượng tích lũy sai số odometry nếu robot di chuyển mà không có vật thể tham chiếu quét liên tục. Nếu không có loop closure hoặc scan matching hiệu quả, dễ dẫn đến bản đồ bị kéo lệch hoặc méo.
- Bố cục hai bên khu vực tương đối giống nhau, có nguy cơ gây nhầm lẫn trong quá trình phát hiện vòng lặp dùng để kiểm tra khả năng vòng lặp của thuật toán.

Tổng quan: Đây là bản đồ có độ khó trung bình cho SLAM:

- Dễ vận hành nhờ hành lang rộng và vật cản đều.
- Có nguy cơ gây lỗi trong hành lang dài nếu không xử lý tốt odometry và phát hiện vòng lặp.

Kết quả bản đồ lập được:



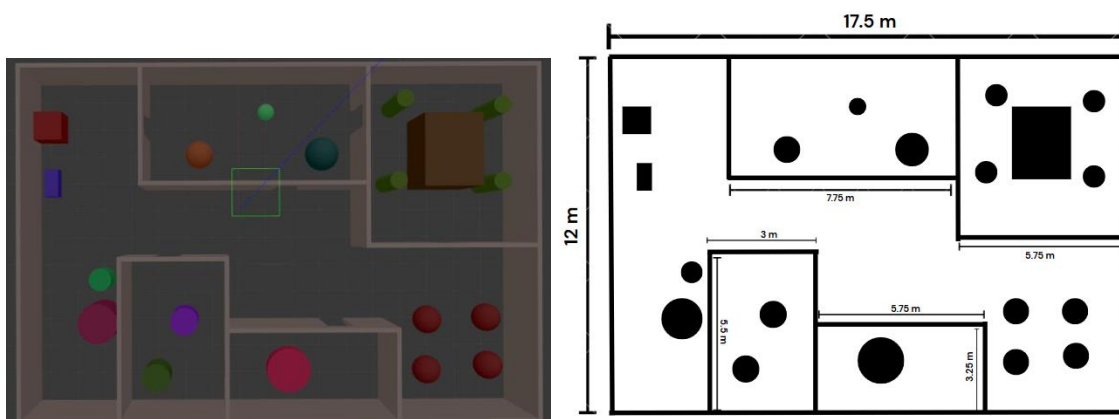
Hình 7.2: Bản đồ lập được

Bản đồ SLAM đã xây dựng được tổng thể môi trường dựa trên thiết kế dạng hành lang chữ U, với hai khu vực hai bên hành lang chứa nhiều vật thể hình tròn và hình vuông.

Tuy gặp các vấn đề gây khó khăn cho slam như cấu trúc tương đối nhưng nhờ điểm mạnh phát hiện vòng lặp của Karto\_slam mà robot vẫn có thể phân biệt được mà không bị nhầm lẫn giữa hai khu vực.

Bản đồ không xuất hiện hiện tượng méo xoắn, vặn lệch toàn cục dấu hiệu cho thấy loop closure đã hoạt động ổn định trong quá trình tối ưu hóa đồ thị pose.

Bản đồ 5.



Hình 8.1: Bản vẽ kỹ thuật bản đồ kiểm thử 5

Bản đồ có kích thước lớn ( $17.5\text{m} \times 12\text{m}$ ), thiết kế thành nhiều phòng nhỏ kết nối với nhau bằng các lối đi và hành lang hẹp. Mỗi phòng bố trí nhiều vật thể hình tròn và hình chữ nhật.

Đặc điểm thuận lợi cho SLAM:

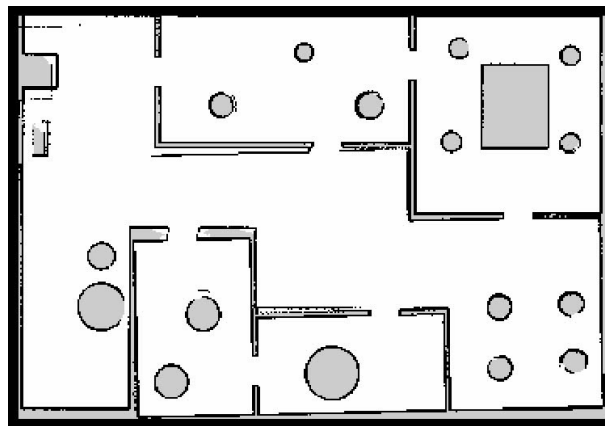
- Các phòng có kích thước và bố cục khác nhau, giúp robot dễ phân biệt khu vực trong quá trình định vị và phát hiện vòng lặp (loop closure).
- Trong mỗi phòng có nhiều vật cản, tạo ra nhiều điểm đặc trưng cho thuật toán scan matching, nâng cao độ ổn định trong việc xây dựng bản đồ.
- Hệ thống bản đồ các phòng rõ ràng, ngăn chia không gian mạch lạc, thuận lợi cho việc tối ưu hóa pose graph khi thực hiện tối ưu toàn cục.

Đặc điểm gây khó khăn cho SLAM:

- Hành lang và lối đi hẹp dễ làm robot bị hạn chế góc quét laser, giảm lượng điểm scan, gây khó khăn cho matching.
- Một số vách ngăn và vật thể lớn (như khối vuông lớn trong phòng góc phải) tạo ra các khu vực blind-spot, có thể khiến robot bỏ lỡ dữ liệu scan hoặc phải quay nhiều lần để thu thập đủ thông tin.
- Độ phức tạp cao do không gian lớn và phức tạp gây khó khăn cho slam trong việc tối ưu hóa đồ thị.

Tổng quan: Đây là bản đồ có độ phức tạp cao đòi hỏi SLAM phải có khả năng quản lý tốt hành lang hẹp, phát hiện vòng lặp thông minh, và tối ưu hóa đồ thị chắc chắn.

Kết quả bản đồ lập được:



*Hình 8.2: Bản đồ lập được*

Bản đồ SLAM đã khôi phục tương đối tốt cấu trúc phức tạp của môi trường thử nghiệm, với nhiều phòng nhỏ, lối đi hẹp và vật thể bố trí dày đặc.

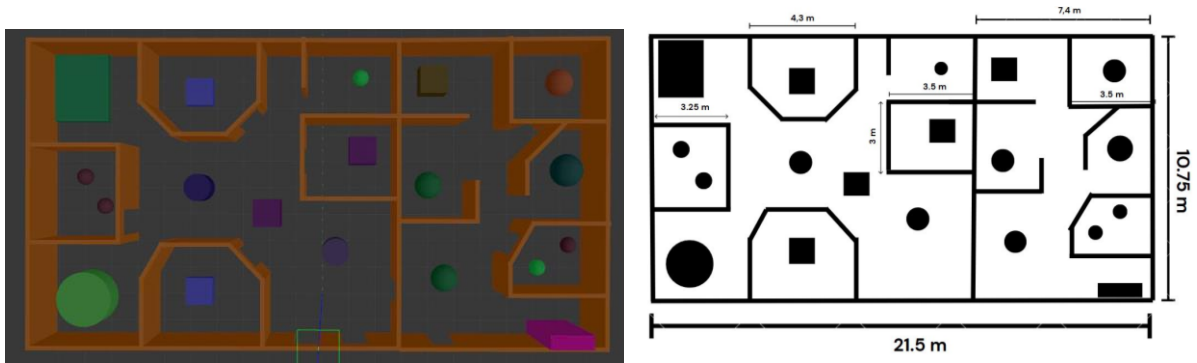
Các phòng, hành lang và các vùng chia nhỏ được tái hiện đúng vị trí và hình dạng, thể hiện rằng quá trình scan matching và tối ưu hóa pose graph đã vận hành ổn định.

Các vật thể hình tròn và hình vuông bên trong các phòng xuất hiện khá đầy đủ và dễ nhận diện, cho thấy dữ liệu từ LiDAR thu thập được chính xác và đồng bộ.

Bản đồ không bị méo xoắn lớn hoặc lệch hình tổng thể, chứng tỏ các vòng lặp (loop closure) đã được phát hiện và tối ưu hóa thành công.

Nhìn chung, bản đồ SLAM thu được đạt chất lượng tốt không bị gặp các lỗi do khó khăn của bản đồ trong quá trình lập bản đồ.

Bản đồ 6.



Hình 9.1: Bản vẽ kỹ thuật bản đồ kiểm thử 6

Bản đồ có kích thước lớn  $21.5\text{m} \times 10.75\text{m}$  là bản đồ có diện tích lớn nhất trong tất cả các bản đồ, được chia thành nhiều phòng nhỏ với hành lang nối giữa, bố trí thêm nhiều vật thể hình tròn và hình vuông, tăng độ phức tạp cho môi trường.

Đặc điểm thuận lợi cho SLAM:

- Các phòng có kích thước khác nhau và bố trí hợp lý giúp robot dễ dàng định vị và phân biệt khu vực.
- Nhiều vật thể lớn nhỏ phân bố khắp nơi làm tăng mật độ thông tin scan, hỗ trợ scan matching chính xác.
- Các phòng có vách nghiêng và vách vuông đan xen, tạo điều kiện tốt để kiểm tra khả năng tối ưu đồ thị trong điều kiện thực tế phức tạp.

Đặc điểm gây khó khăn cho SLAM:

- Các phòng có góc không vuông tiêu chuẩn dễ làm thuật toán scan matching bị sai lệch nhẹ, đặc biệt nếu quét ở xa.
- Các lối đi hẹp kéo dài khiến khả năng phát hiện vòng lặp (loop closure) bị khó hơn nếu tham số loop search không được tinh chỉnh tốt.
- Nếu không kiểm soát tốt tham số scan matching và loop closure, robot dễ bị nhầm lẫn vị trí trong các phòng có sự tương đồng về mặt cấu trúc.

Tổng quan: Đây là bản đồ mô phỏng thực tế phức tạp nhất trong loạt bản đồ thử nghiệm:

- Độ khó cao do yêu cầu kiểm soát tốt matching trong môi trường có nhiều vách nghiêng, hành lang hẹp và khả năng nhầm lẫn khu vực cao.
- Đòi hỏi cao về khả năng tuning tham số scan matching và loop closure, cùng với khả năng tối ưu hóa toàn cục để tránh biến dạng bản đồ.
- Cùng với đó là độ phức tạp của bản đồ.

Kết quả bản đồ lập được:



Hình 9.2: Bản vẽ kỹ thuật bản đồ kiểm thử 6 ở lần quét thứ nhất

Bản đồ thu được có hình dạng tổng thể đúng, nhưng tồn tại nhiều lỗi lớn như:

- Bản đồ bị nghiêng vắn so với hình thiết kế chuẩn ban đầu.
- Một số vách ngăn và tường lớn bị lệch, gây khúc nhẹ hoặc kéo dài không đúng hình học.
- Các khu vực như phòng bên trái và hành lang giữa bị trùng lặp mờ mờ, xuất hiện ghost map.
- Các vật thể nhỏ (hình tròn, hình vuông) không rõ ràng, mờ nhòe và đôi khi lệch nhẹ vị trí.

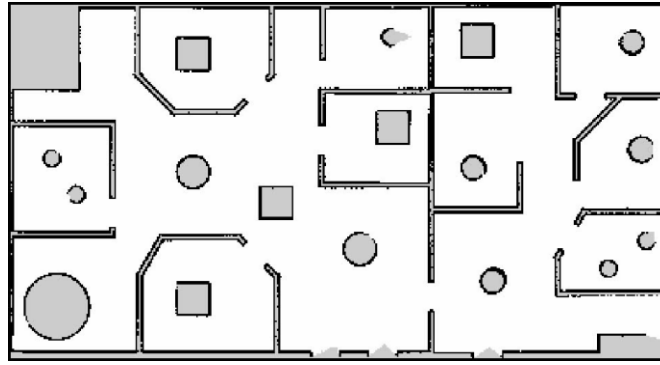
Nguyên nhân gây ra lỗi

- Bản đồ bắt đầu xuất hiện lỗi trong quá trình quét ngay ở khúc cua khi quay lại để quét một phần map còn lại bên trái có thể lý giải do tại đây khi đó đã xảy ra lỗi của sai số odometry dẫn đến sai số tích lũy lớn làm map bị méo mó và vắn vẹo.
- Khi robot di chuyển nhanh, bản đồ cập nhật thưa gây ra mất thông tin scan ở những vùng chuyển tiếp (hành lang, phòng nhỏ) cùng với khoảng cách 0.5m mới tạo node mới làm robot bỏ lỡ những thay đổi nhỏ trong hành lang hoặc phòng nhỏ, thiếu thông tin cho matching.

Tiến hành thay đổi các tham số

- Giảm `minimum_travel_distance` xuống 0.3m để tăng mật độ node, giúp matching và tối ưu tốt hơn.
- Giảm `map_update_interval` xuống 3s để cập nhật bản đồ nhanh hơn, bám sát di chuyển.
- Tăng độ nhạy `loop closure` lên 0.9 để dễ phát hiện vòng lặp trong môi trường nhiều phòng.

Kết quả



Hình 9.3: Bản vẽ kỹ thuật bản đồ kiểm thử 6 ở lần quét thứ hai sau khi điều chỉnh tham số

Tuy bản đồ rất phức tạp và có nhiều đặc điểm nhưng nhờ khả năng mạnh mẽ của mình thuật toán Karto\_slam vẫn có thể xác định được tương đối chính xác bản đồ, tuy nhiên vẫn còn phụ thuộc nhiều vào việc điều chỉnh các tham số cho phù hợp với địa hình của các bản đồ mới có thể cho ra map tối ưu nhất.

## IV. Navigation cho robot.

### A. Kiến trúc hệ thống

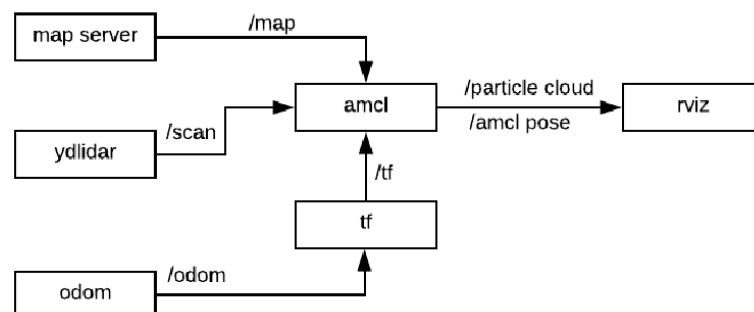
Các thành phần trong navigation stack

Navigation stack trong ROS (thường được gọi là move\_base) là một bộ công cụ mạnh mẽ để điều hướng robot tự động từ điểm A đến điểm B trong một bản đồ đã biết, đồng thời tránh chướng ngại vật.

- move\_base: Đây là node trung tâm của navigation stack, chịu trách nhiệm lập kế hoạch đường đi (path planning) và điều khiển robot di chuyển.
- Các thành phần con:
  - Global Planner: Lập kế hoạch đường đi toàn cục (Global Plan) từ vị trí hiện tại của robot đến mục tiêu, dựa trên bản đồ tĩnh (static\_map). Trong cấu hình của bạn, Global Planner là navfn/NavfnROS (sử dụng thuật toán A\* hoặc Dijkstra để tìm đường đi ngắn nhất).
  - Local Planner: Lập kế hoạch đường đi cục bộ (Local Plan) để điều khiển robot di chuyển, tránh chướng ngại vật động và tĩnh trong thời gian thực. Trong cấu hình của bạn, Local Planner là TrajectoryPlannerROS.
  - Costmaps: Gồm Global Costmap (bản đồ chi phí toàn cục) và Local Costmap (bản đồ chi phí cục bộ), dùng để biểu thị các vùng nguy hiểm mà robot cần tránh.
  - Recovery Behaviors: Khi robot gặp vấn đề (như bị kẹt), move\_base sẽ thử các hành vi khôi phục (recovery behaviors) như xoay tại chỗ hoặc lùi lại.
- Topic chính:

- Input:
  - /map: Bản đồ tĩnh (từ map\_server hoặc SLAM).
  - /scan: Dữ liệu LiDAR để phát hiện chướng ngại vật.
  - /odom: Dữ liệu odometry (vị trí và hướng của robot).
  - /tf: Biến đổi khung tọa độ (TF) giữa các frame như map, odom, base\_link.
  - /move\_base\_simple/goal: Mục tiêu điều hướng (từ RViz hoặc script).
- Output:
  - /cmd\_vel: Lệnh vận tốc để điều khiển robot (linear và angular velocity).
  - /move\_base/global\_costmap/costmap: Global Costmap.
  - /move\_base/local\_costmap/costmap: Local Costmap.
  - /move\_base/TrajectoryPlannerROS/global\_plan: Đường đi toàn cục (Global Plan).
  - /move\_base/TrajectoryPlannerROS/local\_plan: Đường đi cục bộ (Local Plan).
- map\_server: Cung cấp bản đồ tĩnh (static map) cho move\_base sử dụng. Bản đồ thường được lưu dưới dạng file .pgm và .yaml
- AMCL (Adaptive Monte Carlo Localization): Định vị robot trong bản đồ (localization) bằng cách so sánh dữ liệu LiDAR (/scan) với bản đồ tĩnh (/map).
  - Input: /scan, /map, /odom, /tf.
  - Output: /tf (biến đổi từ map -> odom), /amcl\_pose (vị trí ước lượng của robot).

Vai trò: Giúp robot biết vị trí của mình trong bản đồ để điều hướng chính xác.



Hình 10: Cơ chế hoạt động của AMCL trong ROS

- Lập kế hoạch đường đi, hệ thống lập kế hoạch đường đi gồm:

Global Planner: Xây dựng đường đi tổng thể bằng thuật toán A\* dựa trên Global Costmap.

Local Planner: Tính toán vận tốc an toàn theo thuật toán Dynamic Window Approach (DWA) dựa trên Local Costmap.

- **Quy trình:**

Global Planner tìm đường đi tối ưu dựa trên chi phí và heuristic.

Local Planner theo dõi trạng thái robot, tránh vật cản gần, duy trì tiến tới mục tiêu.

Xuất lệnh vận tốc /cmd\_vel để điều khiển robot.

- Thuật toán dò đường trong môi trường mê cung hoạt động như sau:

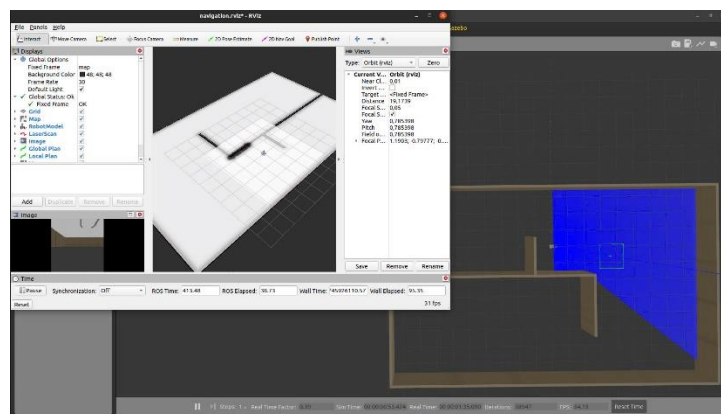
Thu thập khoảng cách bằng cảm biến LaserScan và vị trí qua Odometry.

Xử lý các khoảng cách phía trước, trái, phải bằng các hàm riêng.

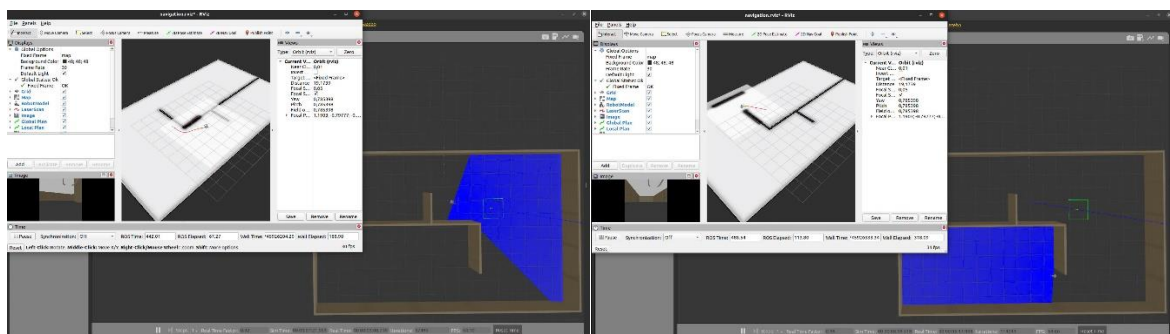
Cập nhật liên tục vị trí robot qua odom\_callback().

Tính toán vận tốc tuyến tính và góc quay, gửi lệnh điều khiển qua /cmd\_vel.

## Kết quả Navigation

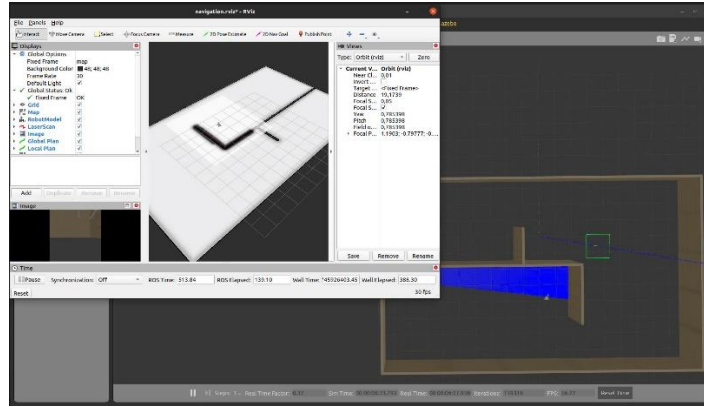


Hình 11.1: Khởi chạy Navigation với map đã được tạo bởi slam



Hình 11.2: Sau khi xác định Pose\_goal thì robot bắt đầu xác định đường đi và di chuyển





*Hình 11.3: Robot sau khi tới nơi*

Robot đã thành công di chuyển đến được vị trí chỉ định mà không gặp các lỗi va chạm hay sai đường. Tuy nhiên đây mới chỉ là thử nghiệm ở một bản đồ tương đối đơn giản, không có vật cản và các chướng ngại vật, cần thử nghiệm trong các bản đồ phức tạp hơn để kiểm thử khả năng tìm đường và xử lý tình huống của robot.

## V. Thảo luận và đánh giá

Dự án đã thành công trong việc triển khai và đánh giá thuật toán Karto SLAM trên nền tảng ROS Noetic, sử dụng mô hình robot mô phỏng trong Gazebo và cảm biến LiDAR 2D. Quá trình thử nghiệm qua 6 bản đồ với độ phức tạp tăng dần đã cung cấp những hiểu biết sâu sắc về hiệu năng và đặc tính của Karto SLAM.

Hiệu năng của Karto SLAM:

- **Điểm mạnh:** Karto SLAM thể hiện khả năng xây dựng bản đồ chính xác cao trong các môi trường có cấu trúc rõ ràng và không quá phức tạp (Bản đồ 1, 2, 4, 5). Điểm mạnh nổi bật là khả năng phát hiện và xử lý vòng lặp (loop closure) hiệu quả, như được chứng minh trong Bản đồ 4 (cấu trúc đối xứng) và các bản đồ khác sau khi tối ưu. Cơ chế tối ưu hóa đồ thị dựa trên Sparse Pose Adjustment (SPA) giúp giảm thiểu đáng kể sai số tích lũy, tạo ra các bản đồ nhất quán về mặt hình học. Việc tận dụng cả dữ liệu odometry và scan matching giúp tăng cường độ chính xác so với các thuật toán chỉ dựa vào scan matching.
- **Thách thức và Hạn chế:**
  - **Độ nhạy với tham số:** Kết quả thử nghiệm trên Bản đồ 3 và Bản đồ 6 cho thấy Karto SLAM rất nhạy cảm với việc cấu hình tham số. Các giá trị mặc định hoặc không phù hợp có thể dẫn đến lỗi nghiêm trọng như méo bản đồ, xoay lệch, hoặc tạo "ghost map" khi môi trường trở nên phức tạp (hành lang hẹp, nhiều góc khuất, cấu trúc lặp lại tương đối).
  - **Ảnh hưởng của môi trường:** Các đặc điểm như hành lang hẹp, góc khuất (blind spots), và bề mặt cong (vật thể tròn) có thể làm giảm chất lượng scan matching do hạn chế góc quét hoặc thiếu đặc trưng rõ ràng. Môi trường lớn và

phức tạp (Bản đồ 6) cũng làm tăng nguy cơ sai số tích lũy từ odometry trước khi vòng lặp được phát hiện và tối ưu.

- Phụ thuộc vào Odometry: Mặc dù Karto SLAM sử dụng scan matching, nó vẫn dựa vào odometry làm ước tính ban đầu. Sai số odometry lớn (như có thể xảy ra khi di chuyển nhanh hoặc quay tại chỗ trong môi trường ít đặc trưng) có thể ảnh hưởng tiêu cực đến quá trình xây dựng bản đồ, đặc biệt là trước khi có đủ ràng buộc từ loop closure.

**Khả năng ứng dụng Navigation:** Thử nghiệm Navigation cơ bản (Hình 11.1-11.3) sử dụng bản đồ được tạo bởi Karto SLAM cho thấy kết quả khả quan. Robot có thể định vị (sử dụng AMCL) và lập kế hoạch đường đi (sử dụng move\_base với global/local planner) để di chuyển đến vị trí đích thành công trên bản đồ tương đối đơn giản. Điều này khẳng định chất lượng bản đồ do Karto SLAM tạo ra là đủ tốt cho các tác vụ điều hướng tự động cơ bản. Tuy nhiên, cần có những thử nghiệm sâu hơn trên các bản đồ phức tạp hơn và trong các tình huống có vật cản động để đánh giá toàn diện hơn.

**Đánh giá chung và Hướng phát triển:**

- **Đánh giá:** Dự án đã đạt được các mục tiêu đề ra, bao gồm triển khai Karto SLAM, tìm hiểu nguyên lý, thử nghiệm trên các bản đồ đa dạng và đánh giá kết quả. Quá trình này cung cấp kinh nghiệm thực tế quý báu về việc sử dụng và tinh chỉnh một thuật toán SLAM phổ biến trong ROS. Karto SLAM chứng tỏ là một lựa chọn mạnh mẽ cho việc lập bản đồ 2D trong nhà, đặc biệt khi yêu cầu độ chính xác cao và có khả năng xử lý vòng lặp tốt, miễn là các tham số được cấu hình phù hợp.
- **Hướng phát triển:**
  - Thử nghiệm trên robot thật: Triển khai trên nền tảng robot vật lý để đánh giá hiệu năng trong điều kiện thực tế với nhiễu cảm biến và sai số odometry thực.
  - So sánh với các thuật toán SLAM khác: Thực hiện so sánh Karto SLAM với các thuật toán khác như GMapping, Cartographer, Hector SLAM trên cùng bộ dữ liệu và môi trường để có cái nhìn toàn diện về ưu nhược điểm của từng phương pháp.
  - Xử lý môi trường động: Nghiên cứu tích hợp các phương pháp phát hiện và loại bỏ vật cản động khỏi bản đồ.
  - Navigation nâng cao: Thực hiện các bài kiểm tra navigation phức tạp hơn trên các bản đồ khó, bao gồm việc xử lý các tình huống bất ngờ hoặc lỗi định vị.

## Trả lời câu hỏi

### Trả lời câu hỏi từ nhóm 1

Câu 1: Ràng buộc từ scanmatching là quá trình so khớp dữ liệu từ lượt quét laser hiện tại với các lượt quét trước đó hoặc một phần bản đồ đã xây dựng và (Loop Closure Detection): là so sánh lượt quét hiện tại với các lượt quét trong quá khứ có khác gì nhau (trước đó và quá khứ)?

- Ràng buộc từ Scan Matching: Là quá trình so khớp lượt quét hiện tại với dữ liệu gần đây.
- Phát hiện Vòng lặp (Loop Closure Detection): Là quá trình so sánh lượt quét hiện tại với dữ liệu trong lịch sử.

Sự khác biệt chính giữa "lượt quét trước đó" (trong Scan Matching) và "lượt quét trong quá khứ" (trong Loop Closure) nằm ở thời gian/không gian và mục đích:

- Scan Matching (So khớp cục bộ):
  - Phạm vi: "Lượt quét trước đó" ở đây thường chỉ lượt quét ngay liền trước (t-1) hoặc một số ít các lượt quét rất gần đây (t-k, với k nhỏ), hoặc một bản đồ cục bộ (local map) được xây dựng từ các lượt quét gần đây đó. Việc so khớp này diễn ra liên tục khi robot di chuyển.
  - Mục đích: Mục đích chính là ước tính chuyển động tương đối chính xác giữa tư thế hiện tại và tư thế ngay trước đó hoặc trong một cửa sổ thời gian/không gian rất ngắn. Nó giúp hiệu chỉnh sai số tức thời của odometry và liên kết các tư thế liên tiếp một cách chính xác cục bộ. Kết quả là một ràng buộc giữa các nút rất gần nhau trong đồ thị.
- Phát hiện Vòng lặp (Loop Closure Detection - So khớp toàn cục):
  - Phạm vi: "Lượt quét trong quá khứ" ở đây ám chỉ các lượt quét có thể đã được thực hiện từ rất lâu trước đó và ở những vị trí địa lý có thể trùng lặp với vị trí hiện tại, không nhất thiết phải gần về mặt thời gian hay thứ tự trong chuỗi di chuyển. Thuật toán cần một cơ chế để tìm kiếm hiệu quả trong toàn bộ hoặc một phần lớn lịch sử các lượt quét/tư thế đã lưu trữ.
  - Mục đích: Mục đích chính là nhận diện việc robot đã quay trở lại một địa điểm đã từng đi qua (place recognition). Khi một vòng lặp được phát hiện và xác nhận, một ràng buộc mới sẽ được tạo ra nối nút hiện tại với một nút rất cũ trong đồ thị. Ràng buộc này cực kỳ quan trọng để hiệu chỉnh sai số tích lũy toàn cục (global drift) đã xảy ra trong suốt quá trình di chuyển dài.

Tóm lại:

- Scan Matching: So khớp với cái ngay gần đây/trước đó à Ước tính chuyển động cục bộ à Ràng buộc cục bộ.
- Loop Closure: So khớp với cái có thể rất cũ/xa trong quá khứ à Nhận diện sự quay lại vị trí cũ à Ràng buộc toàn cục.

Câu 2: Tối ưu hóa ở đây là min edg nghĩa là sao?

Tối ưu hóa đồ thị trong SLAM min edge:

Trong các thuật toán SLAM dạng đồ thị (Graph-based SLAM) như Karto SLAM, Cartographer, GTSAM..., robot sẽ:

- Đại diện các vị trí robot bằng nút (node).
- Đại diện các mối liên hệ giữa các lần đo (odometry, scan matching, loop closure) bằng cạnh (edge) nối các nút.

Mỗi cạnh (edge) chính là một ràng buộc thể hiện giữa 2 vị trí robot: "Theo dữ liệu đo, robot từ node A đến node B di chuyển một khoảng cách cụ thể và góc quay cụ thể."

- Vị trí tương đối thực tế giữa 2 nút A và B (theo đo đạc thực tế),
- Và vị trí tương đối được tính từ pose hiện tại của robot (theo quỹ đạo giả định trong đồ thị).
- Tối ưu hóa là quá trình điều chỉnh vị trí và hướng của tất cả các nút (poses) sao cho:
  - Các cạnh (edges) trong đồ thị gần đúng nhất với dữ liệu đo đạc thực tế.
  - Nghĩa là giảm thiểu sai lệch giữa dự đoán và thực tế tại tất cả các cạnh.

Tóm lại:

- Mỗi cạnh = Một ràng buộc giữa 2 lần đo.
- Tối ưu hóa = Điều chỉnh toàn bộ bản đồ sao cho tổng sai số các cạnh là nhỏ nhất.

VD:

- Nếu robot đi vòng quanh một căn phòng và quay lại điểm xuất phát:
  - Nếu không tối ưu hóa: robot nghĩ mình đang lệch vài mét.
  - Sau khi tối ưu hóa robot điều chỉnh lại toàn bộ quỹ đạo sao cho đầu và cuối khớp nhau phòng trở nên kín và chính xác.

# Tài liệu tham khảo

[1] Grisetti, G., Kummerle, R., Stachniss, C., & Burgard, W. (2010). *A Tutorial on Graph-Based SLAM*. IEEE Intelligent Transportation Systems Magazine, 2(4), 31–43.

<https://doi.org/10.1109/MITS.2010.939925>

[2] Thrun, S., Burgard, W., & Fox, D. (2005). *Probabilistic Robotics*. MIT Press.  
(Chương 7: Monte Carlo Localization, Chương 9: Graph-Based SLAM)

[3] ROS Wiki - Adaptive Monte Carlo Localization (AMCL).

<http://wiki.ros.org/amcl>

[4] ROS Wiki - slam\_karto package.

[http://wiki.ros.org/slam\\_karto](http://wiki.ros.org/slam_karto)

[5] ROS Wiki - Navigation Stack Overview.

<http://wiki.ros.org/navigation>

[6] Kohlbrecher, S., Meyer, J., von Stryk, O., & Klingauf, U. (2011). *A Flexible and Scalable SLAM System with Full 3D Motion Estimation*. Proceedings of the IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR).

[7] Kümmerle, R., Grisetti, G., Strasdat, H., Konolige, K., & Burgard, W. (2011). *g2o: A General Framework for Graph Optimization*. Proceedings of the IEEE International Conference on Robotics and Automation (ICRA).

[8] OpenSLAM.org - Resources and implementations for SLAM algorithms.