

AMR SLAM Đa Tầng Qua Thang Máy

RD Odometry

10/07/2026

Mục lục

Tổng quan	2
Phần 1: Nguyên lý AMR SLAM di chuyển qua thang máy	2
1.1 Kiến trúc tổng quan	2
1.2 Cách biểu diễn bản đồ đa tầng	2
A. Nhiều bản đồ độc lập (phổ biến nhất)	2
B. Một bản đồ 3D / multi-layer	3
C. Topological graph (đồ thị)	3
1.3 Quy trình điển hình (State Machine)	3
Các trạng thái chi tiết	3
1.4 Thách thức SLAM / Localization trong thang máy	3
Vấn đề cốt lõi	3
Cách xử lý thực tế	3
Vai trò của Odometry	4
1.5 Tích hợp thang máy (Elevator Integration)	4
Mức độ tích hợp	4
Tín hiệu thường cần	4
1.6 Global planning xuyên tầng	4
1.7 Relocalization khi sang tầng mới	4
1.8 Luồng dữ liệu tổng thể	5
1.9 Các vấn đề thực tế cần lưu ý	5
Phần 2: Chức năng thang máy trong điều hướng AMR đa tầng	5
2.1 Vai trò vật lý vs vai trò logic	5
2.2 Chức năng theo góc nhìn hệ thống	6
A. Vận chuyển (Transport)	6
B. Cửa và vùng truy cập (Access control)	6
C. Đồng bộ với robot (Synchronization)	6
D. Tài nguyên dùng chung (Resource management)	6
2.3 Interface: Robot ↔ Thang	6
Robot → Thang (Commands)	6
Thang → Robot (Status / Events)	7
2.4 Các chế độ vận hành thang với robot	7
2.5 Chức năng thang tại từng giai đoạn mission	7
Giai đoạn 1: Trước cửa thang (tầng nguồn)	7
Giai đoạn 2: Vào cabin	7
Giai đoạn 3: Trong cabin (đang chạy)	7

Giai đoạn 4: Ra cabin (tầng đích)	8
2.6 Thang như “cạnh” trong đồ thị điều hướng	8
2.7 Chức năng phụ quan trọng	8
2.8 Phân chia trách nhiệm	8
2.9 Ví dụ luồng sự kiện đầy đủ	8
Tóm tắt	9
Ba trụ cột của hệ thống	9
Sáu chức năng cốt lõi của thang máy	9
Kết luận	9

Tổng quan

Tài liệu này tổng hợp nguyên lý điều hướng **AMR (Autonomous Mobile Robot)** sử dụng **SLAM** để di chuyển giữa các tầng thông qua thang máy, và vai trò/chức năng của thang máy trong hệ thống.

Đây là bài toán **điều hướng đa tầng (multi-floor navigation)** — không chỉ là SLAM thuần túy. SLAM giải quyết “robot đang ở đâu trên bản đồ”, còn thang máy thêm lớp **điều phối hệ thống tòa nhà + chuyển đổi ngữ cảnh giữa các tầng**.

Phần 1: Nguyên lý AMR SLAM di chuyển qua thang máy

1.1 Kiến trúc tổng quan

Các thành phần chính trong hệ thống:

Thành phần	Vai trò
SLAM / Localization	Định vị trên bản đồ từng tầng
Global planner	Lập kế hoạch đường đi xuyên tầng
Elevator interface	Gọi thang, mở cửa, giữ cửa, nhận trạng thái
Mission / State machine	Điều phối toàn bộ quy trình lên/xuống tầng
Map management	Quản lý nhiều bản đồ, chuyển tầng

Luồng tương tác:

AMR + SLAM → Navigation / Mission → Elevator Manager

↓

Elevator Controller / BMS

Map Switch / Relocalization

1.2 Cách biểu diễn bản đồ đa tầng

A. Nhiều bản đồ độc lập (phổ biến nhất)

- Mỗi tầng một map riêng (2D occupancy grid hoặc graph).
- Các map **không nối liền tục về mặt hình học** — chỉ liên kết qua **elevator node**.
- Ví dụ: Floor_1.map, Floor_2.map, mỗi map có điểm elevator_A_in, elevator_A_out.

B. Một bản đồ 3D / multi-layer

- Một map chứa nhiều “layer” (mỗi layer = một tầng).
- Phù hợp khi dùng SLAM 3D hoặc hệ thống thương mại (MiR, Omron, Geek+...).

C. Topological graph (đồ thị)

- **Node** = điểm quan trọng (phòng, thang máy, cửa).
- **Edge** = đường đi hoặc **kết nối thang máy giữa các tầng**.
- Global planner chạy trên graph, local planner chạy trên map 2D từng tầng.

1.3 Quy trình điển hình (State Machine)

[Tầng nguồn]	[Thang máy]	[Tầng đích]
Navigate --> Wait at door --> Call elevator --> Enter --> Ride -->		
Align in cabin --> Door close --> Moving --> Door open --> Exit -->		
Relocalize on new floor --> Navigate to goal		

Các trạng thái chi tiết

1. **Navigate to elevator waiting point** — Robot đi đến vị trí chờ trước cửa thang.
2. **Request elevator** — Gửi yêu cầu qua API/BMS: tầng đích, ID thang, ưu tiên.
3. **Wait for arrival** — Chờ thang đến đúng tầng, cửa mở, cabin trống.
4. **Enter elevator** — Local navigation vào cabin (line following, QR marker, hoặc map nhỏ trong cabin).
5. **Inside cabin — localization bị “mù”** — Điểm then chốt kỹ thuật (xem mục 1.4).
6. **Ride to target floor** — Robot đứng yên; hệ thống theo dõi trạng thái thang.
7. **Exit elevator** — Ra khỏi cabin, đến điểm relocalization đã định sẵn.
8. **Relocalize on new floor** — Xác định lại vị trí trên map tầng mới.
9. **Continue mission** — Điều hướng bình thường trên tầng đích.

1.4 Thách thức SLAM / Localization trong thang máy

Vấn đề cốt lõi

Trong cabin, môi trường **thay đổi liên tục** (cửa đóng/mở, cabin di chuyển, người vào ra). LiDAR scan matching dễ **drift** hoặc **nhảy pose sai**.

Cách xử lý thực tế

Giai đoạn	Chiến lược
Trước khi vào	Localization bình thường (AMCL, Cartographer, SLAM Toolbox...)
Trong cabin	Tắt hoặc “đóng băng” SLAM ; chỉ dùng odometry ngắn hoặc IMU
Khi ra cabin	Relocalization bằng scan matching với map tầng mới
Backup	QR / AprilTag / UWB tại cửa thang hai đầu

Vai trò của Odometry

Trong cabin, **wheel odometry + IMU** giữ ước lượng tương đối ngắn (vài mét). Tuy nhiên odometry **không đủ** để biết đã lên tầng nào — cần **tín hiệu từ thang máy** (floor reached).

Luồng điển hình:

SLAM (tin cậy) → Odom only (trong thang) → SLAM relocalize (tầng mới)

1.5 Tích hợp thang máy (Elevator Integration)

Mức độ tích hợp

Mức 1 — Passive (đơn giản)

- Robot chỉ chờ cửa mở (sensor hoặc camera), không điều khiển thang.
- Phù hợp thang công cộng, khó triển khai công nghiệp.

Mức 2 — API với BMS / EMS (phổ biến)

- Giao thức: Modbus, BACnet, OPC-UA, REST API, MQTT.
- Robot gửi: call(floor=5), hold_door(), release().
- Thang trả: current_floor, door_open, moving, fault.

Mức 3 — Dedicated robot elevator

- Thang riêng cho robot (Kone, Otis, Schindler có giải pháp robot elevator).
- Có **robot mode**: giữ cửa lâu hơn, điều khiển từng bước.

Tín hiệu thường cần

Robot → Elevator: call_floor, destination_floor, hold_door, cancel

Elevator → Robot: current_floor, door_status, ready, in_service, fault_code

1.6 Global planning xuyên tầng

Planner không tính đường liên tục 3D. Thường là **2 tầng lập kế hoạch**:

1. Topological planner (đồ thị):

Room_A (F1) → Elevator_1_wait (F1) → [elevator_edge] → Elevator_1_exit (F3) → Room_B (F3)

2. Local planner (từng tầng): DWA, TEB, Nav2 trên map 2D của tầng đó.

Elevator edge là “cạnh ảo”: chi phí = thời gian chờ thang + thời gian di chuyển + rủi ro lỗi.

1.7 Relocalization khi sang tầng mới

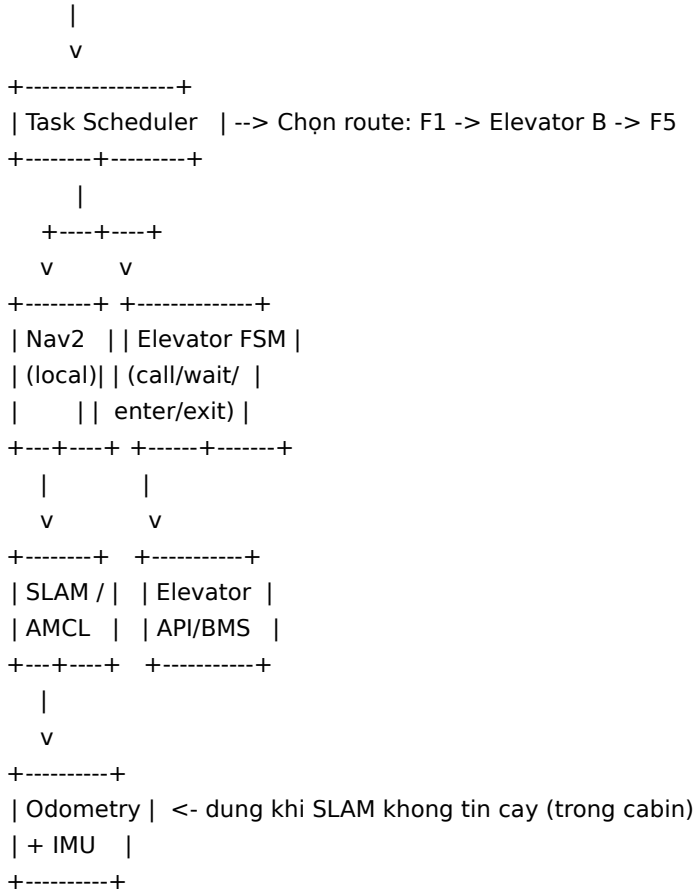
Sau khi ra khỏi thang, robot **không biết chính xác pose** trên map tầng mới (drift trong cabin).

Phương pháp	Mô tả
Scan matching	So khớp LiDAR với map tầng mới (AMCL global localization)
Known exit pose	Điểm ra cố định; initial pose gần đúng
Fiducial marker	QR/AprilTag tại cửa thang
Feature matching	Đặc trưng quanh khu vực cửa thang

Quy trình an toàn: robot ra đến **relocalization zone** → dừng → xác nhận pose (confidence > ngưỡng) → mới tiếp tục.

1.8 Luồng dữ liệu tổng thể

Mission: "Deliver to Room 502 (Floor 5)"



1.9 Các vấn đề thực tế cần lưu ý

1. **An toàn:** E-stop, phát hiện người trong cabin, không chen khi cửa sắp đóng.
2. **Đồng bộ thời gian:** Timestamp map, odometry và trạng thái thang phải nhất quán.
3. **Nhiều robot:** Queue thang, tránh hai robot cùng gọi một cabin.
4. **Map maintenance:** Thay đổi nội thất từng tầng → cập nhật map riêng.
5. **Khe cửa thang / ngưỡng:** Robot phải vượt qua được; thường cần map chi tiết vùng cửa.
6. **Mất mạng:** Timeout khi gọi thang; retry và fallback.
7. **Chuẩn hóa tọa độ:** Mỗi tầng một hệ tọa độ; chỉ nối qua elevator node, không cộng trực tiếp tọa độ giữa các tầng.

Phần 2: Chức năng thang máy trong điều hướng AMR đa tầng

2.1 Vai trò vật lý vs vai trò logic

Lớp	Thang máy làm gì
Vật lý	Chuyển robot từ tầng A sang tầng B
Logic (navigation)	Cạnh nối giữa hai map độc lập (Floor_A ↔ Floor_B)
Điều phối	Tài nguyên dùng chung — nhiều robot có thể tranh nhau
An toàn	Vùng nguy hiểm: cửa, khe, cabin chuyển động

Với SLAM: **thang máy là ranh giới mà localization không còn đáng tin**. Robot “nhảy” từ một không gian định vị được sang không gian khác, còn thang đảm nhiệm phần ở giữa.

2.2 Chức năng theo góc nhìn hệ thống

A. Vận chuyển (Transport)

Chức năng cơ bản: đưa cabin từ tầng nguồn đến tầng đích.

Robot **không tự biết** đã lên tầng nào nếu không có tín hiệu từ thang (current_floor, arrived_at_destination).

B. Cửa và vùng truy cập (Access control)

Thang quyết định **khi nào robot được vào/ra**:

- Cửa mở đủ lâu để robot vào (thường cần **hold door**)
- Cabin dừng đúng tầng, không lệch sàn quá ngưỡng
- Không đóng cửa khi có vật cản (light curtain, cảm biến cửa)

C. Đồng bộ với robot (Synchronization)

Robot FSM		Elevator FSM
-----		-----
GO_TO_ELEVATOR		IDLE
WAIT_CALL	-->	RECEIVE_CALL
WAIT_DOOR_OPEN	<--	ARRIVED + DOOR_OPEN
ENTERING	<->	DOOR_HELD
INSIDE	-->	GO_TO_DEST_FLOOR
WAIT_EXIT	<--	ARRIVED + DOOR_OPEN
EXITING	<->	DOOR_HELD
RELOCALIZE		RELEASE / IDLE

D. Tài nguyên dùng chung (Resource management)

- **Queue** yêu cầu
- **Reservation** (giữ thang cho robot)
- **Priority** (robot giao hàng vs robot dọn phòng)

2.3 Interface: Robot ↔ Thang

Robot → Thang (Commands)

Lệnh	Mục đích
call_elevator(floor)	Gọi thang đến tầng robot đang đứng
request_ride(dest_floor)	Đặt đích sau khi đã vào cabin
hold_door()	Giữ cửa mở khi robot đang vào/ra
release_door()	Cho phép đóng cửa
cancel()	Hủy khi nhiệm vụ thất bại
ping / heartbeat	Giữ session, phát hiện mất kết nối

Thang → Robot (Status / Events)

Tín hiệu	Robot dùng để
current_floor	Biết thang đang ở đâu
door_open / door_closed	Quyết định vào/ra
moving / stopped	Biết cabin đang chạy hay dừng
at_destination	Xác nhận đã đến tầng đích
ready_for_robot	Cabin sẵn sàng (trống, không lỗi)
fault / maintenance	Dừng mission, chọn thang khác
passenger_present	Có người — có thể chờ hoặc hủy

2.4 Các chế độ vận hành thang với robot

Chế độ	Mô tả
Thang công cộng	Robot chỉ quan sát, khó triển khai công nghiệp
Robot mode	API với BMS, giữ cửa lâu, ưu tiên cabin trống — phổ biến nhất
Thang chuyên dụng	Chỉ robot dùng, điều khiển hoàn toàn qua API

2.5 Chức năng thang tại từng giai đoạn mission

Giai đoạn 1: Trước cửa thang (tầng nguồn)

Thang: Nhận call, di chuyển cabin, mở cửa, báo ready.

Robot: Đi đến elevator_wait_point, chờ door_open + current_floor == my_floor.

Giai đoạn 2: Vào cabin

Thang: hold_door trong khi robot vào.

Robot: Tắt/hạ độ tin cậy SLAM, dùng odometry ngắn, gửi request_ride(dest_floor) khi vào xong.

Giai đoạn 3: Trong cabin (đang chạy)

Thang: Báo moving, current_floor, at_destination + door_open.

Robot: Đứng yên, **không cố SLAM**, chờ tín hiệu thang.

Giai đoạn 4: Ra cabin (tầng đích)

Thang: Giữ cửa mở cho đến khi robot báo exited.

Robot: Ra đến relocalization zone, load map tầng đích, relocalize, sau đó release_door.

2.6 Thang như “cạnh” trong đồ thị điều hướng

Floor 1: [Room] --- [Elev1_Wait] =====ELEVATOR===== [Elev1_Exit]

|

Floor 3: [Room] --- [Elev1_Wait] =====ELEVATOR===== [Elev1_Exit]

|

Floor 5: [Room] --- [Elev1_Wait] =====ELEVATOR===== [Elev1_Exit]

Thuộc tính của cạnh thang:

- **Cost** = thời gian chờ + thời gian chạy + độ tin cậy
- **Constraint** = chỉ đi được khi elevator.available == true
- **Precondition** = robot tại wait_point, thang ready
- **Postcondition** = robot tại exit_point tầng đích, đã relocalize

2.7 Chức năng phụ quan trọng

- **Timeout & retry** — Gọi thang quá lâu → đổi thang hoặc hủy mission
- **Interlock an toàn** — Robot trong cabin, thang fault, lệch inside_pose
- **Multi-robot coordination** — Queue, gán elevator_id, tránh tranh cabin
- **Ghi log / audit** — Debug khi SLAM lệch sau khi ra thang

2.8 Phân chia trách nhiệm

Việc	Ai làm
Biết vị trí trên hành lang từng tầng	Robot (SLAM/AMCL)
Lập đường trong tầng	Robot (local planner)
Vào/ra cabin không va chạm	Robot (local nav + odometry)
Relocalize sau khi ra	Robot
Chọn thang nào trong tòa nhà	Fleet / mission manager
Mở cửa, chạy tầng, báo trạng thái	Thang (qua BMS/API)

2.9 Ví dụ luồng sự kiện đầy đủ

1. Mission: F2 → F5 qua Elevator B
2. Robot nav đến Elev_B_wait (F2) [SLAM hoạt động]
3. Robot → Elev: call(2)
4. Elev → Robot: moving_to_2
5. Elev → Robot: arrived, door_open
6. Robot → Elev: hold_door
7. Robot nav vào cabin (2m) [odom chính, SLAM phụ]
8. Robot → Elev: entered, ride_to(5)
9. Robot → Elev: release_door (cho đóng)
10. Elev: moving 2→5

11. Elev → Robot: floor=3,4... (optional)
 12. Elev → Robot: arrived_5, door_open
 13. Robot → Elev: hold_door
 14. Robot nav ra + relocalize (F5) [SLAM bật lại]
 15. Robot → Elev: exited, release_door
 16. Robot nav đến đích trên map F5
-

Tóm tắt

Ba trụ cột của hệ thống

1. **Perception / Localization** — biết vị trí trên từng tầng
2. **Building integration** — điều khiển và đọc trạng thái thang máy
3. **Mission orchestration** — quy trình vào/ra thang và chuyển map

Sáu chức năng cốt lõi của thang máy

1. **Transport** — chuyển tầng vật lý
2. **Gateway** — nối hai không gian/map SLAM tách biệt
3. **Access** — kiểm soát cửa và thời điểm vào/ra
4. **Signaling** — cung cấp trạng thái mà SLAM không suy ra được
5. **Resource** — tài nguyên dùng chung cần đặt chỗ và xếp hàng
6. **Safety envelope** — vùng vận hành có rủi ro cao, cần interlock

Kết luận

AMR multi-floor không phải một SLAM map khổng lồ 3D, mà là nhiều map 2D + đồ thị kết nối qua thang máy + state machine điều phối thang + relocalization khi đổi tầng.

Thang máy là **cầu nối có điều kiện** giữa các map — robot SLAM điều hướng trong từng tầng, còn thang đảm nhiệm **chuyển tầng + đồng bộ trạng thái + kiểm soát cửa**; localization được **tạm ngưng trong cabin** và **khôi phục sau khi ra**.